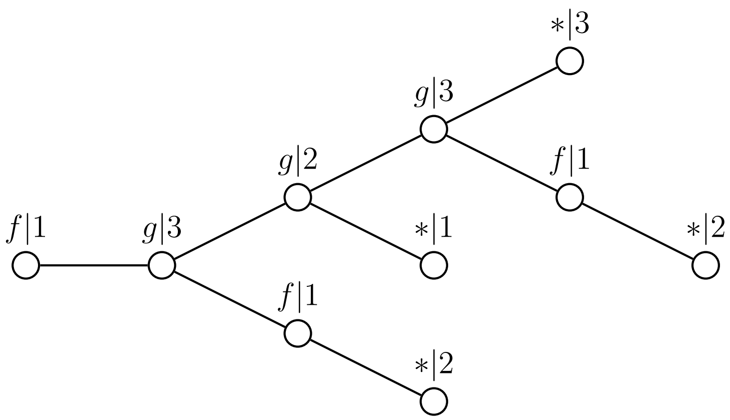
Christian Pech

Kleene-Type Results for Weighted Tree-Automata



DISSERTATION

Kleene-Type Results for Weighted Tree-Automata

Dissertation

zur Erlangung des akademischen Grades

Doktor rerum naturalium

(Dr. rer. nat.)

vorgelegt

der Fakultät Mathematik und Naturwissenschaften

Technischen Universität Dresden

von

Dipl.-Math. Christian Pech

geboren am 22. November 1973 in Dresden

Gutachter : **Prof. Dr. Reinhard Pöschel, TU-Dresden**
Prof. Dr. Manfred Droste, TU-Dresden
Prof. Dr. Zoltán Ésik, University of Szeged

Eingereicht am: **14. April 2003**

Tag der Disputation: **18. August 2003**

Acknowledgments

First I would like to thank to my supervisor Prof. Reinhard Pöschel. He raised my attention to the close connection of algebraic automata-theory with the theory coalgebras. This insight was very influential for the style of this thesis and the methods used therein.

I am grateful to Prof. Manfred Droste. He got me on the track of weighted automata. The initial ideas for this thesis were born in his automata-theory seminar.

I am indebted to Prof. Heiko Vogler for his helpful advises during the preparation of this thesis.

For discussions and hints about content and style of this thesis as well as for proof-reading different preliminary drafts I would like to thank Dietrich Kuske, Dragan Mašulović and Jan Rothe.

I am very grateful to Deutsche Forschungsgemeinschaft (DFG) for supporting my studies in the PhD-program GK334 "Specification of Discrete Processes and Systems of Processes by Operational Models and Logics". The open and interdisciplinary atmosphere of this program provided ideal conditions for research.

Contents

Deutsche Einleitung

Die Theorie der formalen Potenzreihen und der formalen Baumreihen sowie der dazugehörigen gewichteten Automaten ziehen ihre Motivation und ihre Fragen sowohl aus der angewandten und theoretischen Informatik als auch aus der Mathematik. Anwendungen umfassen Bildkompression (Culik und Kari [14], Hafner [31], Katritzke [34], Jiang, Litow und de Vel [33]) und die Übersetzung natürlicher Sprache zu Text (Mohri [42], [43], Buchsbaum, Giancarlo und Westbrook [11]). Aus Sicht der theoretischen Informatik und der Mathematik sind formale Potenz- und Baumreihen vor Allem als Verallgemeinerung der formalen Sprachen bzw. Baumsprachen interessant. Ein wesentliches Teilgebiet der algebraischen Kombinatorik beschäftigt sich mit formalen Potenzreihen über Körpern und Ringen ([3]). Formale Potenzreihen über idempotenten Semiringen finden Anwendungen auf dem Gebiet des Operations Research (cf. [26, 13, 15, 30]).

Natürlich sind die Gründe, sich mit gewichteten Automaten und den ihnen zugeordneten formalen Potenz- bzw. Baumreihen zu beschäftigen, nicht nur rein pragmatischer Natur—den oben genannten Anwendungsgebieten neue Werkzeuge zur Verfügung zu stellen. Vielmehr geht es auch darum, das Verständnis der alten Resultate zu verbessern und tiefere theoretische Zusammenhänge zu erkennen. Exemplarisch für diese These ist das Theorem von Kleene [35] über die Koinzidenz der Klassen der erkennbaren und der rationalen formalen Sprachen. Zunächst wurde es von Schützenberger [47] auf den Fall der formalen Potenzreihen erweitert. Bemerkenswert am Satz von Schützenberger ist seine totale Unabhängigkeit von der Wahl des Koeffizientensemiringes. Thatcher und Wright [49] verallgemeinerten später das Kleene-Theorem auf formale Baumsprachen. Aufbauend auf dem Satz von Schützenberger haben Droste und Gastin [16] einen Kleene-artigen Satz für formale Potenzreihen auf Spurmonoiden bewiesen.

Der nächste logische Schritt war nun natürlich die Verallgemeinerung dieses Resultats auf formale Baumreihen. Kuich demonstrierte einen Kleene-artigen Satz für erkennbare formale Baumreihen in [36]. Er benutzte dabei Fixpunkttheorie für cpos von formalen Baumreihen und zahlte dafür einen verhältnismäßig hohen Preis; Sein Satz gilt nur unter der Einschränkung, daß der Koeffizientensemiring kommutativ, vollständig, natürlich geordnet und stetig ist. In [9] bewies Bozapalidis ein entsprechendes Theorem für ω-wohladditive, natürlich geordnete Semiringe. Schließlich wurden alle diese Resultate von Bloom und Ésik in [6] auf den Fall der Conway-Semiringe verallgemeinert. Allerdings scheint die Einschränkung auf Conway-Semiringe ist immer noch zu stark zu sein, betrachtet man die Allgemeinheit des Satzes von Schützenberger.

Eine weitere Version des Kleene-Theorems für erkennbare Baumreihen wurde von Droste und Vogler in [18] vorgestellt. Ihr Beweis ist elementar und benutzt insbesondere keine Fixpunktheorie. Dadurch benötigen sie nun nurmehr die Kommutativität und Idempotenz des Koeffizientensemiringes. Insbesondere kann hier auf die Existenz von bestimmten unendlichen Summen gänzlich verzichtet werden.

In [2] charakterisieren Berstel und Reutenauer die erkennbaren formalen Baumreihen als Komponenten von eindeutigen Lösungen gewisser Gleichungssysteme über Polynomen. Die Arbeit betrachtet ausschließlich formale Baumreihen über kommutativen Körpern. Eine Charakterisierung der erkennbaren Reihen durch rationale Ausdrücke wird nicht angegeben. Deshalb gehen wir davon aus, daß ein Kleene-artiger Satz für formale Baumreihen über Körpern bis zu diesem Zeitpunkt nicht existiert.

Ziel dieser Arbeit ist es nun, alle bisher erhaltenen Resultate zu Kleene-artigen Sätzen für formale Baumreihen zu erweitern mit dem Ziel, eine ähnliche Allgemeinheit wie im Satz von Schützenberger zu erreichen. Die Hauptschwierigkeit besteht dabei darin, daß die Menge der formalen Baumreihen über einem beliebigen Semiring einfach zu wenig Struktur besitzt, um einen direkten Angriff zu stützen. Insbesondere fehlt ihr die Ordnungsstruktur (bzw. die kategorielle Struktur), die es uns sonst erlaubt kleinste (oder initiale) Fixpunkte zu betrachten. Deshalb ist zu Beginn noch nicht einmal klar, wie die verschiedenen Iterationsoperatoren, die aus der Welt der Baumsprachen bekannt sind, für formale Baumreihen zu definieren sind. Ad hoc Definitionen der Iterationsoperatoren sind zwar denkbar, haben aber immer den Nachteil, keine legitimen Verallgemeinerungen der bekannten Operationen (vgl. [49, 20]) zu sein.

Wir lösen dieses Dilemma, indem wir den Kleene-Satz für formale Baumreihen auf einem anderen, höheren semantischen Niveau beweisen und dieses Resultat nachher mit einer natürlichen semantischen Abstraktionsabbildung auf die Ebene der formalen Baumreihen transportieren. Die eben erwähnte höhere Semantik modellieren wir, indem wir den Begriff der gewichteten Baumsprache einführen. Das sind Multimengen von Bäumen (über einer gegebenen Signatur), deren Knoten mit Gewichten aus einem gegebenen Semiring versehen sind. Gewichtete Automaten erkennen nun gewichtete Baumsprachen anstatt von formalen Baumreihen. Mit einem natürlichen Homomorphiebegriff bildet die Klasse der gewichteten Baumsprachen eine vollständige und covollständige Kategorie mit initialen und terminalen Objekten. Die verschiedenen rationalen Operationen wie Topkatenation, a-Produkt etc. auf Baumsprachen können zu Funktoren der Kategorie der gewichteten Baumsprachen verallgemeinert werden. Es zeigt sich, daß alle so gewonnenen Funktoren sich sehr wohlverhalten—sie bewahren monos und alle oder zumindest alle gerichteten Colimites. Somit können auch die Iterationsoperationen als initiale Algebrenträger gewisser Funktoren eingeführt werden. Diese Funktoren sind wiederum direkte Verallgemeinerungen der bekannten Funktionen, die zur Definition der Iteration von formalen Baumsprachen betrachtet werden. Nocheinmal betonen wir, daß die hohe Regularität dieser Umgebung vollkommen unabhängig von der Wahl des Koeffizientensemiringes ist.

An diesem Punkt eröffnen sich uns zwei Möglichkeiten zum Beweis eines Kleene-Satzes für gewichtete Baumsprachen. Die eine folgt den Arbeiten von Kuich, Bozapalidis, Bloom and Ésik (cf. [36, 9, 22, 6]) und benutzt Fixpunkttheorie. Die andere folgt klassischen Arbeiten zum Kleene-Satz bei denen direkte Konstruktionen von rationalen Ausdrücken aus Automaten und vice versa angegeben werden (vgl. [49, 27], siehe auch [18]). Der erste Weg hat den Vorteil, algebraisch und dadurch recht elegant zu sein. Insbesondere kann hier kann unser Kleene-Satz aus einem abstrakteren Satz von Bloom und Ésik [6] über Conway-Grovetheorien gefolgert werden kann. Diese Eleganz wird aber mit geringer Transparenz und einem hohen begrifflichen und technischen Aufwand erkauft. Aus diesem Grund haben wir uns in dieser Arbeit für den zweiten Weg entschieden. Das heißt, unser Beweis wird weitestgehend elementar sein und auf Automatenkonstruktionen beruhen.

Nachdem in *Abschnitt 1* Grundbegriffe wie "gewichtete Bäume" und "Rangmonoide" definiert und die für uns wesentlichen Eigenschaften bewiesen werden, führt *Abschnitt 2* den in dieser Arbeit zentralen Begriff der gewichteten Baumsprache ein. Die Kategorie WTL_Σ der gewichteten Baumsprachen wird untersucht und die bekannten rationalen Operationen aus der Welt der formalen Baumsprachen (vgl. [49, 20]) werden auf gewichtete Baumsprachen verallgemeinert. Dabei wird jede Operation zunächst auf recht abstrakte kategorielle Weise definiert, um sie einfach auf ihre kategoriellen Eigenschaften hin untersuchen zu können. Danach wird für die meisten Operationen noch eine intuitivere Konstruktion angegeben und deren Gleichwertigkeit zur kategoriellen Definition bewiesen. Es zeigt sich, daß WTL_Σ eine vollständige und covollständige Kategorie mit initialen und terminalen Objekten ist in der der Produktfunktor beliebige Colimites bewahrt. Diese Eigenschaften sind ein kategorielles Pendant zu vollständigen vollständig distributiven Verbänden. Darüber hinaus stellt sich heraus, daß alle von uns eingeführten rationalen Operationen gerichtete Colimites bewahren. Das ist eine kategorielle Eigenschaft, die die Stetigkeit von Funktionen auf vollständigen partiellen Ordnungen verallgemeinert. Rationale Operationen, die auf formalen Baumsprachen distributiv über der Vereinigung sind, bewahren in ihrer abstrakten Version beliebige Colimites.

In *Abschnitt 3* wird auf konventionelle Art definiert, was ein gewichteter Baumautomat (WTA) ist. Im Zusammenhang damit wird erklärt, was eine erkennbare gewichtete Baumsprache ist. *Abschnitt 4* führt den Begriff des schwachen WTA (wWTA) ein, in denen zusätzlich "stille" Transitionen zugelassen werden. Diese stillen Transitionen entsprechen in etwa den ε-Transitionen in der klassischen Automatentheorie. Anders als dort erkennen wWTAs eine strikt größere Klasse von gewichteten Baumsprachen als die WTAs. Deshalb werden Sprachen, die von einem wWTA erkannt werden "schwach erkennbar" genannt. Es wird genau charakterisiert, wann eine schwach erkennbare gewichtete Baumsprache erkennbar ist und unter welcher Bedingung aus einem wWTA die stillen Transitionen eliminierbar sind (ohne die erkannte Sprache zu verändern). Danach werden die rationalen Operationen für wWTAs definiert und die entsprechenden Verträglichkeitsbeweise geführt. Eine sofortige Konsequenz ist, daß die schwach erkennbaren gewichteten Baumsprachen unter den rationalen Operationen abgeschlossen sind. Dies gilt nicht für die erkennbaren gewichteten Baumsprachen. Insbesondere sind erkennbare gewichtete Baumsprachen im Allgemeinen nicht unter den Iterationsoperationen abgeschlossen. Wir geben notwendige und hinreichende Bedingungen an, unter denen die Iteration einer erkennbaren Sprachen wieder erkennbar ist.

Abschnitt 5 enthält das erste Hauptergebnis der Arbeit. Zunächst wird die Klasse der rekursiven Ausdrücke definiert (wir haben sie nicht "rationale Ausdrücke" genannt, da sie keine Verallgemeinerung des klassischen Begriffs der rationalen Ausdrücke darstellen). Theorem 5.22 charakterisiert die schwach erkennbaren gewichteten Baumsprachen mittels rekursiver Ausdrücke und Theorem 5.23 charakterisiert die erkennbaren Sprachen mit Hilfe der echten rekursiven Ausdrücke.

In *Abschnitt 6* werden verschiedene Klassen rationaler Ausdrücken eingeführt. Theorem 6.4 charakterisiert die schwach erkennbaren und Theorem 6.9 charakterisiert die erkennbaren gewichteten Baumsprachen mit Hilfe von rationalen Ausdrücken. Erst jetzt, in *Abschnitt 7*, werden Resultate für formale Baumreihen bewiesen. Zunächst wird ein Zusammenhang zwischen den gewichteten Baumsprachen und den formalen Baumreihen hergestellt. Nicht jede gewichtete Baumsprache erlaubt eine Interpretation als formale Baumreihe, sondern nur die sogenannten finitären gewichteten Baumsprachen (vgl. 2.32). Danach werden, wie zu erwarten, die rationalen Operationen für formale Baumreihen definiert. Eine Verträglichkeit mit den bisherigen Definitionen der rationalen Operationen auf gewichteten Baumsprachen ist dabei im Allgemeinen nur dann zu erzielen, wenn von einem kommutativen Koeffizientensemiring ausgegangen wird. Eine entsprechende Einschränkung muß in Theorem 7.25 gemacht werden, wo die erkennbaren formalen Baumreihen durch rationale Ausdrücke charakterisiert werden. Dieses Theorem kann man als das Hauptresultat dieser Arbeit ansehen. Es stellt die gewünschte Verallgemeinerung der bisherigen Resultate zu Kleene-artigen Sätzen für formale Baumreihen dar.

Die letzten drei Abschnitte sollen die fixpunkttheoretischen Aspekte von gewichteten Baumsprachen bzw. formalen Baumreihen beleuchten. Zum einen wollen wir dadurch unser Resultat mit den Resultaten von Kuich [36], Bozapalidis[9] and Bloom und Ésik [6] vergleichen. Zum anderen wollen wir ein Resultat von Berstel und Reutenauer [2] über die Charakterisierung von erkennbaren formalen Baumreihen durch eindeutige Lösungen gewisser Gleichungssysteme verallgemeinern.

Abschnitt 8 führt in die grundlegenden Begriffe der Fixpunkttheorie ein. In *Abschnitt 9* assoziieren wir mit jeder gewichteten Baumsprache einen Funktor auf $\mathsf{WTL}_{\Sigma(X)}$, der gerichtete Colimites bewahrt. Diese Zuordnung ist injektiv bis auf Isomorphie. Wir zeigen, daß die so gewonnenen Funktoren eine Iterationstheorie bestimmen. Weiterhin charakterisieren wir schwach erkennbare und erkennbare Sprachen mit Hilfe von Normalbeschreibungen—einer Art abstrakter Automaten in Iterationstheorien. Schließlich folgern wir einen fixpunkttheoretischen Kleene-Satz für schwach erkennbare gewichtete Baumsprachen á la Bloom-Ésik (vgl. 9.21).

Abschnitt 10 beschäftigt sich mit der Fixpunktheorie auf formalen Baumreihen über kommutativen Semiringen. Zunächst führen wir in üblicher Weise die Theorie der formalen Baumreihen ein und zeigen, daß es sich dabei um eine sogenannte Grove-Theorie handelt (siehe auch [6]). Wir beweisen, daß diese Grove-Theorie eine partielle Iterationstheorie ist. Das Ideal, auf dem die Fixpunktoperation definiert ist, besteht aus allen vollkommen quasiregulären Morphismen der Theorie. Weiterhin charakterisieren wir erkennbare formale Baumreihen durch quasireguläre Normalbeschreibungen. Schließlich zeigen wir, daß es genau eine Möglichkeit gibt, auf den quasiregulären Morphismen eine Fixpunktoperation zu definieren und daß diese Fixpunktoperation die Erkennbarkeit von Morphismen bewahrt. Das erweitert und verallgemeinert das Resultat von Berstel und Reutenauer [2], daß jedes echte System linearer Gleichungen über Polynomen genau eine Lösung hat, das diese Lösung erkennbar ist und daß darüber hinaus jede erkennbare formale Baumreihe eine Komponente der eindeutigen Lösung eines geeigneten linearen Gleichungssystems über Polynomen entspricht.

Introduction

The theory of formal power-series, tree-series, and their corresponding weighted automata draws its motivation and questions from applied and theoretical computer science as well as mathematics. Applications comprise image compression (Culik & Kari [14], Hafner [31], Katritzke [34], Jiang, Litow & de Vel [33]) and the translation of natural language to text (Mohri [42], [43], Buchsbaum, Giancarlo & Westbrook [11]). From the perspective of theoretical computer science and mathematics, formal power- and tree-series are interesting especially as generalizations of formal languages and tree-languages. An important subarea of algebraic combinatorics studies the formal power-series over fields and rings ([3]). They play a role theire mainly as generating functions of certain combinatorial objects. Formal power-series over idempotent semirings also found a special interest because they have applications in operations research (see e.g. [26, 13, 15, 30]). Last but not least let us mention that formal tree-series also play a role as tool for syntax-directed semantics of programming languages. There in particular tree-series transducers are of interest (cf. [21, 24, 25])

Of course the reasons to work with weighted automata and their formal power- and tree-series is not only of purely pragmatical nature—to provide tools for the above mentioned applications. Rather we would also like to get to a better understanding of the known results about formal languages and tree-languages to see deeper theoretical connections. Exemplary for this thesis is Kleene's theorem [35] about the coincidence of the classes of rational and recognizable formal languages. At first this result was generalized by Schützenberger [47] to formal power series. It is remarkable that Schützenberger's result does not depend at all on the underlying semiring. A bit later Thatcher & Wright [49] extended Kleene's theorem to formal tree-languages and recently in [16] Droste and Gastin generalized Schützenberger's theorem to formal power-series over trace-monoids (this result is also independent from the coefficient semiring).

Now, the next logical step was to unify Schützenberger's theorem about formal power series and Thatcher's & Wright's theorem about formal tree-languages. In [36] Kuich demonstrated a Kleene-type theorem for recognizable formal tree-series. In his proof he used fixed point theory on complete partial orders (short: cpo) and had to pay a rather high price for this: His Theorem only holds under the assumption that the coefficient semiring is commutative, complete, naturally ordered and continuous. Another partial result of this kind was obtained by Bozapalidis [9]. Here the coefficient semiring must be well ω-additive, commutative and naturally ordered. Recently Bloom & Ésik proved a Kleene-type theorem for commutative Conway-semirings. Hence they generalized Kuich's and Bozapalidis' results. At the same time Droste & Vogler [18] published a Kleene-type result for idempotent, commutative semirings. While Kuich, Bozapalidis, Bloom & Ésik used mainly fixed point theoretical methods, the proof by Droste and Vogler is elementary and bases mainly on automata-theoretic constructions.

In [2] Berstel and Reutenauer show among other results, that the recognizable formal tree-series arise as components of unique solution of certain systems of equations over polynomials. The formal tree-series that they consider take their coefficients out of a (commutative) field. A characterization of recognizable series by rational expressions is not provided by them. Therefore we consider a Kleene-type theorem for formal tree-series over fields as nonexistent at the moment.

The goal of this thesis is to find a common generalization to all above mentioned Kleene-type results for formal tree-series in order to reach a similar generality like in Schützenberger's theorem. The main difficulty is, that the set of formal power series over an arbitrary semiring does carry too little structure in order to support a direct attack. In particular it is lacking the order structure that was so essential in the previous results and that allowed to consider least fixed points of certain operations. Such tools are of particular importance when we would like to define the iteration of formal tree-series. Ad hoc-definitions for iteration are conceivable but they always have the disadvantage not to be legitimate generalizations of the known operations (cf. [49, 20]).

We resolve the dilemma by proving the Kleene-type theorem for formal tree-series on another, higher semantical level and by transporting this result later on down again to formal tree-series using a natural abstraction map. The just mentioned higher semantics we model by weighted tree-languages. These are multisets of trees whose nodes are equipped with weights from the given semiring. Now weighted automata recognize weighted tree-languages instead of formal tree-series. The weighted tree-languages, together with a natural homomorphism concept, form a complete and cocomplete category with initial and final objects. The known rational operations on formal tree-languages (cf. [49]) like a-product, topcatenation etc. can be generalized to weighted tree-languages naturally. There they are functors that turn out to be in some sense very well-behaved—all of them preserve directed colimits and if some operation on formal tree-languages was distributive over the union of languages, then its weighted pendant preserves arbitrary colimits. This regularity makes it easy to introduce the iteration operations on weighted tree-languages as initial algebra carriers of certain functors that preserve directed colimits. These functors are again direct generalizations of the known operations that are used to define the iteration of formal tree-languages. We emphasize that the high regularity of this environment is completely independent from the choice of the coefficient semiring.

At this point we see two options for the proof of a Kleene-type result for weighted tree-languages. One of them follows the papers by Kuich, Bozapalidis, Bloom & Ésik (cf. [36, 9, 22, 6]) and uses fixed point theory. The other one follows more classical automata-theoretic methods that give direct constructions of rational expressions from automata and vice versa (cf. [49, 27], see also [18]). The first approach has the indisputable advantage to be algebraical and to be therefore rather elegant. However, this elegance is payed for with a very high level of abstraction and therefore with rather low transparency. Therefore we preferred the second option. This way our proofs can be elementary and mostly self-contained and in fact the automata-constructions that we do, give a nice intuition about the rational operations.

After the introduction of elementary notions like “weighted trees” and “ranked monoids” in *Section 1* and the examination of their basic properties, we introduce in *Section 2* the central notion of weighted tree-languages. The category WTL_Σ of the weighted tree-languages is examined and the known rational operations from the world of formal tree-languages (cf. [49, 20]) are generalized to weighted tree-languages. In doing so all operations are at first defined in a rather abstract, categorial way as functors in order to allow an easy examination of their categorial properties (like preservation of colimits). After doing so, for most of the operations we provide more vivid constructions and show their equivalence to the original definitions. It turns out that WTL_Σ is a complete and cocomplete category with initial and terminal objects. These properties are a categorial pendant to complete lattices[1]. Moreover we find that all our rational operations preserve directed colimits. This is the categorial analog of continuity of functions on complete partial orders. Those rational operations on formal tree-languages that are distributive over union, turn out to preserve arbitrary colimits in their weighted version.

In *Section 3* we give a conventional definition of weighted tree-automata (WTAs). In connection with this we explain the concept of recognizable weighted tree-language. *Section 4* generalizes WTAs to weak weighted tree-automata (wWTAs) by introducing “silent transitions”. These silent transitions correspond to the ε-transitions in classical automata-theory. In contrast with classical automata-theory, wWTAs recognize a strictly wider class of weighted tree-languages than WTAs. Therefore the weighted tree-languages that are recognized by a wWTA are called “weakly recognizable”. We characterize completely when a weakly recognizable weighted tree-language is recognizable and under which conditions the silent transitions in a wWTA may be eliminated (without altering the recognized weighted tree-language). After that the rational operations are defined on wWTAs and their compatibility with the corresponding operations on weighted tree-languages is proved. As an immediate consequence we obtain that weakly recognizable weighted tree-languages are closed with respect to the rational operations. This does not hold for the recognizable weighted tree-languages since they are in general not closed with respect to the iteration operations. We give necessary and sufficient conditions, when the iteration of a recognizable weighted tree-language is again recognizable.

Section 5 contains the first main result of the dissertation. At first we define the set of fp-expressions (we did not call them “rational expressions” since they are not a proper generalization of the classical rational expressions for formal tree-languages or for formal languages). Theorem 5.22 characterizes the weakly recognizable weighted tree-languages by fp-expressions and Theorem 5.23 gives a similar characterization of the recognizable weighted tree-languages (these theorems are related to the Kleene-type theorems by Kuich [36] and Ésik & Kuich [22]; in par-

[1] In fact it can be shown that the product-functor preserves arbitrary colimits in WTL_Σ. This is a categorial pendant of complete distributivity in complete lattices.

ticular they use the same system of expressions).

In *Section 6* we introduce different classes of rational expressions. Theorem 6.4 characterizes the weakly recognizable and Theorem 6.9 characterizes the recognizable weighted tree-languages with the help of rational expressions. Moreover, in Corollaries 6.6 and 6.10 these results are used to characterize (weakly) recognizable weighted tree-languages by certain kinds of rational closures of the finite weighted tree-languages (this is related to the Kleene-type result by Droste & Vogler [18, Thms. 4.1, 5.1]). Only now, in *Section 7*, we prove results for formal tree-series. First we give a connection between weighted tree-languages and formal tree-series. Not every weighted tree-language allows an interpretation as formal tree-series, but only the so called finitary weighted tree-languages do (cf. 2.32). After this, expectedly, the rational operations for formal tree-series are defined. In general, a compatibility with the corresponding operations on weighted tree-languages does only hold if the coefficient semiring is commutative. An according restriction must therefore be made in Theorem 7.25 where the recognizable formal tree-series are characterized by rational expressions. Using this result, the recognizable formal tree-series are characterized as some rational closure of the polynomials in Corollary 7.26 (this generalizes the Kleene-type result by Droste & Vogler [18, Thms. 4.1, 5.1]).

The last three sections shed some light onto the fixed point theoretical aspects of formal tree-languages and formal tree-series. One motivation for this is to give a comparison of our Kleene-type theorem with the results by Kuich [36], Bozapalidis [9] and Bloom & Ésik [6]. On the other hand we would like to generalize a result by Berstel & Reutenauer [2] about the characterization of recognizable formal tree-series by components of unique solutions of certain systems of equations over polynomials.

Section 8 introduces the basic notions from fixed point theory. In *Section 9* we associate with each weighted tree-language a functor on $\mathsf{WTL}_{\Sigma(X)}$ that preserves directed colimits. This association is injective up to isomorphism. We show that these functors determine an iteration theory. Further on, in Propositions 9.19 and 9.20, we characterize the (weakly) recognizable weighted tree-languages by so called (quasiregular) normal descriptions—an abstract automata-model from fixed point theory. Later, in Corollary 9.21 we deduce a fixed point theoretical Kleene-type theorem for weakly recognizable weighted tree-languages á la Bloom & Ésik [6, Thm. 9.4]. Finally, in Corollary 9.23, we deduce from the previous results a characterization of the weakly recognizable scalar arrows in the theory of weighted tree-languages as a rational closure of scalar morphisms corresponding to finite languages (this result is related to the Kleene-type result by Bloom & Ésik [6, Cor. 10.3] and to the one by Bozapalidis [9]).

Section 10 deals with the fixed point theory of formal tree-series. At first we introduce the theory of formal tree-series in the usual way and show once more that we obtain a so called grove-theory (cf. [6]). We introduce a partial fixed point operation on this theory and in Proposition 10.23 we show that this operation fulfills several partial identities. Using this we conclude in Theorem 10.24 that the grove-theory of formal tree-series is in fact a partial iteration theory (in some sense this generalizes a similar result by Bloom & Ésik [6, Cor. 8.17]). The ideal on which the fixed point operation is defined, consists of all fully quasiregular morphisms of the theory. Beyond this, in Theorem 10.28, we characterize the recognizable formal tree-series by quasiregular normal descriptions (this generalizes a similar characterization by Bloom & Ésik [6, Thm. 9.4]). Finally, in Theorem 10.31, we show that there is precisely one way to define a fixed point operation on the quasiregular morphisms and in Corollary 10.26 we observe that this fixed point operation preserves the recognizability of morphisms. All this together augments and generalizes the results by Berstel & Reutenauer [2] and by Bozapalidis [9], that each proper (quasiregular) linear system of equations over polynomials has precisely one solution, that the components of this solution are recognizable formal tree-series and that moreover every recognizable formal tree-series arises as a component of the unique solution of a convenient system of equations. Finally, in Corollary 10.34 we characterize the recognizable scalar morphisms in the theory of formal tree-series by a rational closure (this generalizes a Kleene-type result by Bloom & Ésik [6, Cor. 10.2] and one by Bozapalidis [9]).

Let us mention in the end that this thesis tries to be as self-contained as possible. However, since we use terminology from automata-theory, general algebra, category theory and fixed point theory, it is not feasible to really give a good introduction of each of these fields. Instead we will only give those definitions that we really use. As standard references from automata theory we mention Eilenberg [19] and Kuich & Salomaa [38]. As standard reference for universal algebra we mention Grätzer [28] and Burris & Sankappanavar [12]. For notions from category theory we mention the books by Mac Lane [41], Adámek, Herrlich & Strecker [1] and Borceux [7]. And finally the standard reference for fixed point theory is the book by Bloom & Ésik [4].

1 Trees, Weighted Trees and Ranked Monoids

In this section we are going to give some of the basic notions for this thesis concerning trees and weighted trees. Apart from their definition this includes also the introduction of some operations on trees and weighted trees such as e.g. depth, a-size, a-substitution and their basic properties. Finally we study two different but important algebraic structures that can be introduced on the set of (weighted) trees—(weighted) Σ-algebras, ranked semigroups and ranked monoids.

1.1 Trees, tree-languages. A *ranked alphabet* (or *ranked set*) is is a pair (Σ, rk) where Σ is a set of letters (an alphabet) and $\mathrm{rk} : \Sigma \longrightarrow \mathbb{N}$ assigns to each letter a natural number (its rank)[2]. With $\Sigma^{(n)}$ we denote the set of letters from Σ with rank n. If X is a set with $\Sigma \cap X = \emptyset$, then with $\Sigma(X)$ we will denote the ranked alphabet given by $(\Sigma \cup X, \mathrm{rk}')$ where

$$\mathrm{rk}'(x) = \begin{cases} \mathrm{rk}(x) & x \in \Sigma \\ 0 & \text{otherwise.} \end{cases}$$

The set T_Σ of *trees* over Σ is the least set defined inductively by $\Sigma^{(0)} \subseteq \mathrm{T}_\Sigma$ and if $a \in \Sigma^{(n)}$ and $t_1, \ldots, t_n \in \mathrm{T}_\Sigma$, then $a\langle t_1, \ldots, t_n\rangle \in \mathrm{T}_\Sigma$.

A Σ-*tree-language* is a subset of T_Σ. With TL_Σ we will denote the set of all Σ-treelanguages partially ordered by inclusion.

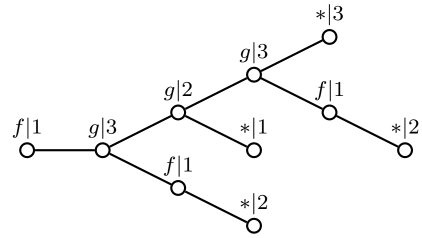
1.2 Semirings. A *semiring* is a quintuple $(K, \oplus, \odot, 0, 1)$ such that $(K, \oplus, 0)$ is a commutative monoid, $(K, \odot, 1)$ is a monoid and such that the following identities hold: $(x \oplus y) \odot z = (x \odot z) \oplus (y \odot z)$, $x \odot (y \oplus z) = (x \odot y) \oplus (x \odot z)$ and $x \odot 0 = 0 \odot x = 0$. As usual, we identify the carrier K of the semiring with the semiring if the operations are clear from the context.

1.3 Weighted trees. Let $(K, \oplus, \odot, 0, 1)$ be a semiring. A *weighted tree* is a tree where each node is provided with an element of K—its weight. More precisely, with the notions from above we define the set WT_Σ of weighted Σ-trees inductively: If $a \in \Sigma^{(0)}$ and $c \in K$ then $[a|c] \in \mathrm{WT}_\Sigma$. If $f \in \Sigma^{(n)}$ and $t_1, \ldots, t_n \in \mathrm{WT}_\Sigma$ and $c \in K$, then $[f|c]\langle t_1, \ldots, t_n\rangle \in \mathrm{WT}_\Sigma$.

1.4 Underlying tree. To each weighted tree t we may associate its *underlying tree* $\mathrm{ut}(t)$ which intuitively is obtained from t by forgetting the weights on each node of t. Technically we define it inductively on the structure of t: For $a \in \Sigma^{(0)}$, $c \in K$ we set $\mathrm{ut}([a|c]) := a$. For $f \in \Sigma^{(n)}$ and $t_1, \ldots, t_n \in \mathrm{WT}_\Sigma$ and $c \in L$ we define $\mathrm{ut}([f|c]\langle t_1, \ldots, t_n\rangle) := f\langle \mathrm{ut}(t_1), \ldots, \mathrm{ut}(t_n)\rangle$.

[2]Throughout this thesis $\mathbb{N}$ denotes the set of all non-negative integers

1.5 Example. Let $K = (\mathbb{N}, +, \cdot, 0, 1)$. Let $\Sigma := \Sigma^{(0)} \cup \Sigma^{(1)} \cup \Sigma^{(2)}$ where $\Sigma^{(0)} = \{*\}$, $\Sigma^{(1)} = \{f\}$ and $\Sigma^{(2)} = \{g\}$. We are going to give a pictorial representation of a weighted tree $t \in \mathrm{WT}_\Sigma$ with underlying tree $f\langle g\langle f\langle *\rangle, g\langle *, g\langle f\langle *\rangle, *\rangle\rangle\rangle\rangle$:



Note that we use the convention that the outputs of each node are ordered counterclockwise starting directly after the input-edge.

1.6 Address-based description of (weighted) trees. In order to have an alternative description of weighted trees, we associate to each node an address from $\mathbb{N}^*$—the set of all formal words that can be formed out of letters from $\mathbb{N}$. Let $t \in \mathrm{WT}_\Sigma$. Then we define $\mathrm{adr}(t) \subseteq \mathbb{N}^*$ inductively by

$$\mathrm{adr}([a|c]) := \{\varepsilon\}, \qquad (a \in \Sigma^{(0)},\ c \in K)$$

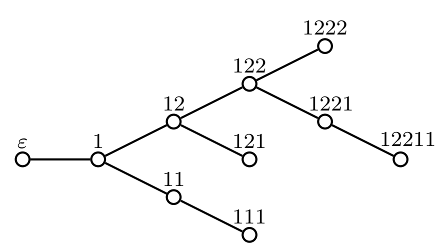
$$\mathrm{adr}([f|c]\langle t_1, \dots, t_n\rangle) := \{\varepsilon\} \cup \bigcup_{i=1}^{n} \{i \cdot w \mid w \in \mathrm{adr}(t_i)\}, \qquad (f \in \Sigma, n = \mathrm{rk}(f),\ c \in K\ \ t_1, \dots, t_n \in \mathrm{WT}_\Sigma).$$

Here ε denotes the empty word and $i \cdot w$ denotes the usual concatenation of the word consisting just of the letter i with the word w. Additionally we define functions $\mathrm{lab}_t : \mathrm{adr}(t) \longrightarrow \Sigma$ and $\mathrm{wt}_t : \mathrm{adr}(t) \longrightarrow K$ according to

$$\begin{aligned} \mathrm{lab}_{[a|c]}(\varepsilon) &:= a & \mathrm{wt}_{[a|c]}(\varepsilon) &:= c \\ \mathrm{lab}_{[f|c]\langle t_1,\dots,t_n\rangle}(\varepsilon) &:= f & \mathrm{wt}_{[f|c]\langle t_1,\dots,t_n\rangle}(\varepsilon) &:= c \\ \mathrm{lab}_{[f|c]\langle t_1,\dots,t_n\rangle}(i \cdot w) &:= \mathrm{lab}_{t_i}(w) & \mathrm{wt}_{[f|c]\langle t_1,\dots,t_n\rangle}(i \cdot w) &:= \mathrm{wt}_{t_i}(w). \end{aligned}$$

Clearly, each $t \in \mathrm{WT}_\Sigma$ is completely determined by the triple $(\mathrm{adr}(t), \mathrm{lab}_t, \mathrm{wt}_t)$. Analogously (non-weighted) trees may be characterized by a pair $(\mathrm{adr}(t), \mathrm{lab}_t)$. These address-based descriptions will be useful in later proofs.

1.7 Example. We give the address-based description of the weighted tree from 1.5:



w	$\text{lab}_t(w)$	$\text{wt}_t(w)$	w	$\text{lab}_t(w)$	$\text{wt}_t(w)$
ε	f	1	121	$*$	1
1	g	3	122	g	3
11	f	1	1221	f	1
12	g	2	1222	$*$	3
111	$*$	2	12211	$*$	2

1.8 a-Size. An important property of weighted trees is their a-size, where $a \in \Sigma^{(0)}$. The *a-size* of a weighted tree t is just the number of nodes that are not labeled by a. To be precise we give its inductive definition: $\text{size}_a([a|c]) := 0$ and $\text{size}_a([f|c]\langle t_1, \dots, t_n\rangle) := 1 + \sum_{i=1}^{n} \text{size}_a(t_i)$. The definition of size_a on T_Σ is analogous. Obviously, for any $t \in \text{WT}_\Sigma : \text{size}_a(t) = \text{size}_a(\text{ut}(t))$.

1.9 Depth. Another useful property for weighted trees is their *depth* (cf. [3, Example 2.1]). This is the length of a maximal descending path in the tree starting from the root. Technically it is defined inductively: $\text{depth}([a|c]) := 0$ for $a \in \Sigma^{(0)}$ and $\text{depth}([f|c]\langle t_1, \dots, t_n\rangle) := 1 + \max(\text{depth}(t_1), \dots, \text{depth}(t_n))$. As usual the definition of the depth of non-weighted trees is analogous.

1.10 Product with scalars on WT_Σ. K acts naturally on WT_Σ from the left according to $d \cdot [a|c] := [a|d \odot c]$ and $d \cdot [f|c]\langle t_1, \dots, t_n\rangle := [f|d \odot c]\langle t_1, \dots, t_n\rangle$. Obviously we have $(c \odot d) \cdot t = c \cdot (d \cdot t)$ for all $c, d \in K$ and $t \in \text{WT}_\Sigma$.

1.11 a-substitution. Next we define the operation of *a-substitution* on weighted trees where $a \in \Sigma^{(0)}$. Given a weighted tree t with n nodes labeled by a and given further weighted trees $t_1, \dots, t_n$. Then $t \circ_a \langle t_1, \dots, t_n\rangle$ is obtained from t by substituting the i-th leaf of t labeled with a (counted from the "left") by t_i thereby adjusting the costs by multiplication with the weight of the root of t_i (cf. Figure 1). The exact definition goes inductively on the structure of t. On the way we also define the a-rank $\text{rk}_a(t)$ of t:

For $c \in K$, we define $\text{rk}_a([a|c]) := 1$ and for $t_1 \in \text{WT}_\Sigma$: $[a|c] \circ_a \langle t_1\rangle := c \cdot t$. For any $b \in \Sigma^{(0)}$ different from a, $\text{rk}_a([b|c]) := 0$ and $[b|c] \circ_a \langle\rangle := [b|c]$. Suppose $f \in \Sigma^{(n)}$ and $t_1, \dots, t_n \in \text{WT}_\Sigma$ with a-ranks $m_1, \dots, m_n$, respectively. Then $\text{rk}_a([f|c]\langle t_1, \dots, t_n\rangle) := m_1 + \dots + m_n$ and for $s_{1,1}, \dots, s_{1,m_1}, \dots, s_{n,m_n} \in \text{WT}_\Sigma$ we define

$$[f|c]\langle t_1, \dots, t_n\rangle \circ_a \langle s_{1,1}, \dots, s_{n,m_n}\rangle := \\ [f|c]\langle t_1 \circ_a \langle s_{1,1}, \dots, s_{1,m_1}\rangle, \dots, t_n \circ_a \langle s_{n,1}, \dots, s_{n,m_n}\rangle\rangle.$$

The definition of a-substitution on trees is completely analogous.

1.12 Lemma. *For $c \in K$ and $t \in \text{WT}_\Sigma$ we have $\text{ut}(c \cdot t) = \text{ut}(t)$. Moreover, for $t \in \text{WT}_\Sigma$ with $\text{rk}_a(t) = n$ and for $s_1, \dots, s_n \in \text{WT}_\Sigma$ we have*

$$\text{ut}(t \circ_a \langle s_1, \dots, s_n\rangle) = \text{ut}(t) \circ_a \langle \text{ut}(s_1), \dots, \text{ut}(s_n)\rangle.$$

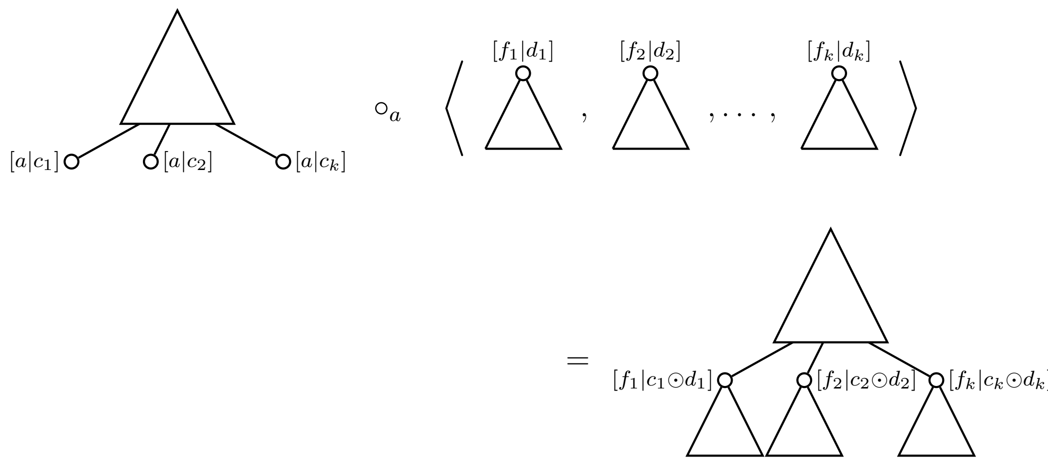


Figure 1: a-substitution

Proof. The first claim follows directly from 1.10.

Further, concerning the second claim, we proceed by induction on the structure of t:

$$\mathrm{ut}([a|c] \circ_a \langle t_1 \rangle) = \mathrm{ut}(c \cdot t_1) = \mathrm{ut}(t_1) = a \circ_a \mathrm{ut}(t_1) = \mathrm{ut}([a|c]) \circ_a \mathrm{ut}(t_1)$$
$$\mathrm{ut}([b|c] \circ_a \langle\rangle) = \mathrm{ut}([b|c]) = b = \mathrm{ut}([b|c]) \circ_a \langle\rangle.$$

$$\begin{aligned}
&\mathrm{ut}([f|c]\langle t_1, \dots, t_n \rangle \circ_a \langle s_{1,1}, \dots, s_{n,m_n} \rangle) \\
&\quad = \mathrm{ut}([f|c]\langle t_1 \circ_a \langle s_{1,1}, \dots, s_{1,m_1} \rangle, \dots, t_n \circ_a \langle s_{n,1}, \dots, s_{n,m_n} \rangle \rangle) \\
&\quad = f\langle \mathrm{ut}(t_1 \circ_a \langle s_{1,1}, \dots, s_{1,m_1} \rangle), \dots, \mathrm{ut}(t_n \circ_a \langle s_{n,1}, \dots, s_{n,m_n} \rangle) \rangle \\
&\quad = f\langle \mathrm{ut}(t_1) \circ_a \langle \mathrm{ut}(s_{1,1}), \dots, \mathrm{ut}(s_{1,m_1}) \rangle, \dots, \\
&\qquad\qquad \mathrm{ut}(t_n) \circ_a \langle \mathrm{ut}(s_{n,1}), \dots, \mathrm{ut}(s_{n,m_n}) \rangle \rangle \\
&\quad = f\langle \mathrm{ut}(t_1), \dots, \mathrm{ut}(t_n) \rangle \circ_a \langle \mathrm{ut}(s_{1,1}), \dots, \mathrm{ut}(s_{n,m_n}) \rangle \\
&\quad = \mathrm{ut}([f|c]\langle t_1, \dots, t_n \rangle) \circ_a \langle \mathrm{ut}(s_{1,1}), \dots, \mathrm{ut}(s_{n,m_n}) \rangle.
\end{aligned}$$

□

1.13 Lemma. *Let* s, $s_1, \dots, s_k \in \mathrm{WT}_\Sigma$ *where* $\mathrm{rk}_a(s) = k$. *Then*

$$\mathrm{size}_a(s \circ_a \langle s_1, \dots, s_k \rangle) = \mathrm{size}_a(s) + \sum_{i=1}^{k} \mathrm{size}_a(s_i).$$

Proof. We proceed by induction on the structure of s:

$$\begin{aligned}
\mathrm{size}_a([a|c] \circ_a \langle s_1 \rangle) &= \mathrm{size}_a(c \cdot s_1) = \mathrm{size}_a(s_1) = 0 + \mathrm{size}_a(s_1) \\
&= \mathrm{size}_a([a|c]) + \mathrm{size}_a(s_1).
\end{aligned}$$

$$
\begin{aligned}
&\mathrm{size}_a([f|c]\langle t_1,\dots,t_n\rangle \circ_a \langle s_{1,1},\dots,s_{n,m_n}\rangle)\\
&\qquad= \mathrm{size}_a([f|c]\langle t_1 \circ_a \langle s_{1,1},\dots,s_{1,m_1}\rangle,\dots,t_n \circ_a \langle s_{n,1},\dots,s_{n,m_n}\rangle\rangle)\\
&\qquad= 1 + \sum_{i=1}^{n}\left(\mathrm{size}_a(t_i) + \sum_{j=1}^{m_i}\mathrm{size}_a(s_{i,j})\right)\\
&\qquad= \mathrm{size}_a([f|c]\langle t_1,\dots,t_n\rangle) + \sum_{i=1}^{n}\sum_{j=1}^{m_i}\mathrm{size}_a(s_{i,j})
\end{aligned}
$$

□

1.14 Corollary. *With the notions from above* $\mathrm{size}_a(s \circ_a \langle s_1,\dots,s_k\rangle) \geq \mathrm{size}_a(s)$. *Equality holds if and only if* $\mathrm{size}_a(s_i) = 0$ *for all* $i = 1,\dots,k$. □

1.15 Algebraic structure of WT_Σ. When an algebraist is given a ranked alphabet (Σ, rk) then his first impulse is to consider it as the signature for a category of Σ-algebras. Then each element of Σ represents the name of an operation of these algebras whose arity is given by rk. For instance boolean algebras are algebras over the signature $\Sigma = \{\wedge, \vee, *, 0, 1\}$ (obeying additional axioms) where $\mathrm{rk}(\wedge) = \mathrm{rk}(\vee) = 2$, $\mathrm{rk}(*) = 1$ and $\mathrm{rk}(0) = \mathrm{rk}(1) = 0$. Indeed, the set T_Σ of Σ-trees is nothing else but a convenient representation of the carrier of the free Σ-algebra.

In order to understand WT_Σ algebraically, we need to consider another class of algebras – the *weighted algebras*. Here a *K-weighted Σ-algebra* is a Σ-algebra $(A, (\hat{f})_{f\in\Sigma})$ on which the semiring K acts from the left. That is, to each element c of K there corresponds a unary operation $c \cdot - : A \longrightarrow A$ where $x \mapsto c \cdot x$ such that $(c \odot d) \cdot x = c \cdot (d \cdot x)$ for all $c, d \in K$, $x \in A$. It is not difficult to see that WT_Σ, together with

$$\hat{f}(t_1,\dots,t_n) := [f|1]\langle t_1,\dots,t_n\rangle$$

and with the action of K from 1.10 forms a weighted Σ-algebra. It is in fact the free K-weighted Σ-algebra. Note that T_Σ may also be considered as a K-weighted Σ-algebra. Here the action of K on the left is trivial. That is $c \cdot t = t$ for all $c \in K$ and $t \in \mathrm{T}_\Sigma$. The mapping $\mathrm{ut} : \mathrm{WT}_\Sigma \longrightarrow \mathrm{T}_\Sigma$ is then just the corresponding initial (and therefore unique) homomorphism of K-weighted sigma-algebras.

Another way to deal with ranked alphabets is to take them as generalization of usual alphabets from formal language theory. To be more precise, every alphabet A corresponds to a so-called *monadic ranked alphabet*. This is a ranked alphabet where each letter has rank 0 or 1. In particular we may choose $\Sigma_A = (A \dot{\cup} \{*\}, \mathrm{rk})$ where $\mathrm{rk}(x) = 1$ if $x \in A$ and $\mathrm{rk}(*) = 0$. Indeed, there is an obvious one to one correspondence between the set of words A^* and the set of trees T_{Σ_A}. There is just one little flaw in this generalization since T_{Σ_A} carries naturally the structure of a Σ-algebra while A^* is usually equipped with the structure of a monoid. Moreover semigroup- and monoid theory play a rather prominent role in formal language theory. Therefore it is natural to ask about an adequate generalization of monoids and semigroups to the case of ranked sets. The structures we will define now are called magmas by Berstel and Reutenauer in [2]. Since this term is used to denote an entirely different class of structures (sets with a binary operation), we invent new names.

1.16 Ranked semigroups. A *ranked semigroup* is a triple $(S, \mathrm{rk}, \circ)$ where (S, rk) is a ranked set and where $\circ = (\circ_i)_{i \in \mathbb{N}}$ is a family of partial composition operations:

$$\circ_i \,:\, S^{(i)} \times S^i \longrightarrow S$$
$$\circ_i \,:\, (f, (g_1, \dots, g_i)) \mapsto f \circ \langle g_1, \dots, g_i \rangle$$

such that

1. $\mathrm{rk}(f \circ \langle g_1, \dots, g_n \rangle) = \sum_{i=1}^{n} \mathrm{rk}(g_i)$,
2. $\forall n, \forall f \in S^{(n)}, \forall m_1, \dots, m_n, \forall g_i \in S^{(m_i)}, \forall h_{i,j} \in S \; (i = 1, \dots, n, \; j = 1, \dots, m_i)$:

$$(f \circ \langle g_1, \dots, g_n \rangle) \circ \langle h_{1,1}, \dots, h_{1,m_1}, \dots, h_{n,1}, \dots, h_{n,m_n} \rangle$$
$$= f \circ \langle g_1 \circ \langle h_{1,1}, \dots, h_{1,m_1} \rangle, \dots, g_n \circ \langle h_{n,1}, \dots, h_{n,m_n} \rangle \rangle$$

1.17 Remark. The second axiom in this definition is called the *superassociativity law*. From the first sight it may look scary. On the other hand it is a direct generalization of the associativity law of semigroups.

1.18 Ranked monoids. A *ranked monoid* is a quadruple $(S, \mathrm{rk}, \circ, \varepsilon)$ where

1. $(S, \mathrm{rk}, \circ)$ is a ranked semigroup,
2. $\varepsilon \in S^{(1)}$,
3. $\forall n \, \forall f \in S^{(n)} \,:\, f \circ \langle \varepsilon, \dots, \varepsilon \rangle = f$,
4. $\forall f \in S \,:\, \varepsilon \circ \langle f \rangle = f$.

1.19 Remark. Note that there is a connection of ranked monoids and ranked semigroups to clone-theory. The operation of linearized composition in clones fulfills all requirements of a composition operation in ranked semigroups. And the identity mapping that is contained in each clone serves as the unit in a ranked monoid. For more information on clones see [44].

1.20 Free ranked monoids. Ranked semigroups and ranked monoids may be considered as special instances of heterogeneous (or many-sorted) algebras. Here the sorts are the natural numbers. This observation allows us to speculate on the rich structure of such algebras without having to develop a theory of ranked semigroups and -monoids ourself. In particular we conclude that for any ranked alphabet (Σ, rk) there is a free ranked monoid freely generated by (Σ, rk). It will be denoted by $(\Sigma, \mathrm{rk})^*$. The carrier of $(\Sigma, \mathrm{rk})^*$ may be constructed as follows: First we extend the alphabet by one letter: $\Sigma' := \Sigma \cup \{\varepsilon\}$. The rank-function is extended to Σ' according to $\mathrm{rk}(\varepsilon) := 0$. Then the carrier of $(\Sigma, \mathrm{rk})^*$ is just $\mathrm{T}_{\Sigma'}$. Moreover, $(\Sigma, \mathrm{rk})^* := (\mathrm{T}_{\Sigma'}, \mathrm{rk}_\varepsilon, \circ_\varepsilon, \varepsilon)$. Where rk_ε and $\circ_\varepsilon$ are defined like in 1.11. Axioms 3 and 4 of ranked monoids are easily verified.

We still need to show superassociativity. We proceed by induction on the structure of the left-most tree: First we note that

$$(\varepsilon \circ_\varepsilon \langle t \rangle) \circ_\varepsilon \langle s_1, \ldots, s_n \rangle = t \circ_\varepsilon \langle s_1, \ldots, s_n \rangle = \varepsilon \circ_\varepsilon \langle t \circ_\varepsilon \langle s_1, \ldots, s_n \rangle \rangle.$$

If the leftmost tree is equal to a for $a \in \Sigma^{(0)}$, then the superassociativity-claim becomes trivial.

The induction step is simple but terrible. Therefore we only dare to give in script style:

$$\begin{aligned}
&(f\langle g_1,\ldots,g_k\rangle \circ_\varepsilon \langle t_{1,1},\ldots,t_{1,l_1},\ldots,t_{k,1},\ldots,t_{k,l_k}\rangle)\circ_\varepsilon\langle s_{1,1,1},\ldots,s_{1,1,m_{1,1}},\ldots,s_{k,l_k,1},\ldots,s_{k,l_k,m_{k,l_k}}\rangle\\
&\quad=(f\langle g_1\circ_\varepsilon\langle t_{1,1},\ldots,t_{1,l_1}\rangle,\ldots,g_k\langle t_{k,1},\ldots,t_{k,l_k}\rangle\rangle)\circ_\varepsilon\langle s_{1,1,1},\ldots,s_{1,1,m_{1,1}},\ldots,s_{k,l_k,1},\ldots,s_{k,l_k,m_{k,l_k}}\rangle\\
&\quad=f\langle(g_1\circ_\varepsilon\langle t_{1,1},\ldots,t_{1,l_1}\rangle)\circ_\varepsilon\langle s_{1,1,1},\ldots,s_{1,l_1,m_{1,l_1}}\rangle,\ldots,(g_k\circ_\varepsilon\langle t_{k,1},\ldots,t_{k,l_k}\rangle)\circ_\varepsilon\langle s_{k,1,1},\ldots,s_{k,l_k,m_{k,l_k}}\rangle\rangle\\
&\quad=f\langle g_1\circ_\varepsilon\langle t_{1,1}\circ_\varepsilon\langle s_{1,1,1},\ldots,s_{1,1,m_{1,1}}\rangle,\ldots,t_{1,l_1}\circ_\varepsilon\langle s_{1,l_1,1},\ldots,s_{1,l_1,m_{1,l_1}}\rangle\rangle,\ldots,g_k\circ_\varepsilon\langle\ldots\rangle\rangle.
\end{aligned}$$

Summing up we have $(\Sigma, \mathrm{rk})^* = (\mathrm{T}_{\Sigma'}, \mathrm{rk}_\varepsilon, \circ_\varepsilon, \varepsilon)$ is a ranked monoid. We can assume that (Σ, rk) is a generating set by using the natural identification $\iota : (\Sigma, \mathrm{rk}) \rightarrowtail (\Sigma, \mathrm{rk})^*$ where $f \mapsto f\langle \varepsilon, \ldots, \varepsilon \rangle$. Later on we will apply this identification implicitly.

It remains to show that (Σ, rk) is a free generating set. Let $(M, \mathrm{rk}, \circ, \varepsilon_M)$ be another ranked monoid and let $\varphi : (\Sigma, \mathrm{rk}) \longrightarrow (M, \mathrm{rk}_M)$ be a rank preserving map. We will show that there is a unique extension $\varphi^\# : (\Sigma, \mathrm{rk})^* \longrightarrow (M, \mathrm{rk}, \circ, \varepsilon_M)$ of φ to a homomorphism of ranked monoids. The proof of this will be done in two steps. First we construct $\varphi^\#$ and show its uniqueness. Then we show that $\varphi^\#$ is a homomorphism of ranked monoids.

Let us extend φ to $\mathrm{T}_{\Sigma'}$ by induction. For $t = \varepsilon$ we must have $\varphi^\#(t) = \varepsilon_M$ and for $t = a \in \Sigma^{(0)}$ there is no other choice but to define $\varphi^\#(t) := \varphi(t)$. Moreover, for $f \in \Sigma^{(n)}$. $\varphi^\#(\iota(f)) := \varphi(f)$ because $\varphi^\#$ should extend φ. Suppose now $t = f\langle t_1, \ldots, t_n \rangle$ for some $f \in \Sigma^{(n)}$. Then $t = \iota(f) \circ_\varepsilon \langle t_1, \ldots, t_n \rangle$. Hence, by induction hypothesis and since $\varphi^\#$ should become a homomorphism, we must define $\varphi^\#(t) := \varphi(f) \circ \langle \varphi^\#(t_1), \ldots, \varphi^\#(t_n) \rangle$. Thus $\varphi^\#$ is welldefined and unique. Clearly, it is also rank preserving.

Now we show that $\varphi^{\#}$ is a homomorphism. That is, we have to show that $\varphi^{\#}(\varepsilon) = \varepsilon_M$ and $\varphi^{\#}(t \circ_\varepsilon \langle t_1, \dots, t_n \rangle) = \varphi^{\#}(t) \circ \langle \varphi(t_1), \dots, \varphi(t_n) \rangle$. The first property is given by the construction of $\varphi^{\#}$. The second property is proved by induction on the structure of t. If $t = \varepsilon$, then $\varphi^{\#}(\varepsilon \circ_\varepsilon \langle t_1 \rangle) = \varphi^{\#}(t_1) = \varepsilon_M \circ \langle \varphi^{\#}(t_1) \rangle = \varphi^{\#}(\varepsilon) \circ \langle \varphi^{\#}(t_1) \rangle$. If $t = a \in \Sigma^{(0)}$, then $\varphi^{\#}(a \circ_\varepsilon \langle \rangle) = \varphi^{\#}(a) = \varphi^{\#}(a) \circ \langle \rangle$. If $t = f\langle s_1, \dots, s_m \rangle$, then

$$\begin{aligned}
&\varphi^{\#}(f\langle s_1, \dots, s_m \rangle \circ_\varepsilon \langle t_{1,1}, \dots, t_{m,n_m} \rangle \\
&\qquad = \varphi^{\#}((\iota(f) \circ_\varepsilon \langle s_1, \dots, s_m \rangle) \circ_\varepsilon \langle t_{1,1}, \dots, t_{m,n_m} \rangle) \\
&\qquad = \varphi^{\#}(\iota(f) \circ_\varepsilon \langle s_i \circ_\varepsilon \langle t_{i,1}, \dots, t_{i,n_i} \rangle_{i=1}^{m} \rangle) \\
&\qquad = \varphi(f) \circ \langle \varphi^{\#}(s_i \circ_\varepsilon \langle t_{i,1}, \dots, t_{i,n_i} \rangle) \rangle_{i=1}^{m} \\
&\qquad = \varphi(f) \circ \langle \varphi^{\#}(s_i) \circ \langle \varphi^{\#}(t_{i,1}), \dots, \varphi^{\#}(t_{i,n_i}) \rangle \rangle_{i=1}^{m} \quad \text{by induction hypothesis} \\
&\qquad = \varphi(f) \circ \langle \varphi^{\#}(s_1), \dots, \varphi^{\#}(s_m) \rangle \circ \langle \varphi^{\#}(t_{1,1}), \dots, \varphi^{\#}(t_{m,n_m}) \rangle \\
&\qquad = \varphi^{\#}(f\langle s_1, \dots, s_m \rangle) \circ \langle \varphi^{\#}(t_{1,1}), \dots, \varphi^{\#}(t_{m,n_m}) \rangle .
\end{aligned}$$

Hence $\varphi^{\#}$ is a homomorphism. Summing up (Σ, rk) is a free generating set of $(\Sigma, rk)^*$ (under the identification ι). Hence $(\Sigma, \mathrm{rk})^*$ is indeed free, freely generated by (Σ, rk).

1.21 Remarks. With the construction of the free ranked monoid above we see easily that for given (Σ, rk) and $a \in \Sigma^{(0)}$ the quadruple $(\mathrm{T}_\Sigma, \mathrm{rk}_a, \circ_a, a)$ forms a ranked monoid. It is in fact freely generated by $(\Sigma \setminus \{a\}, \mathrm{rk}')$ where rk' is the restriction of rk to $\Sigma \setminus \{a\}$. Similarly we can prove that also $(\mathrm{WT}_\Sigma, \mathrm{rk}_a, \circ_a, [a|1])$ is a ranked monoid. It is not freely generated as a ranked monoid by $(\Sigma \setminus \{a\}, \mathrm{rk}')$ but as a *weighted ranked monoid* it is. There a weighted ranked monoid is a ranked monoid $(M, \mathrm{rk}, \circ, \varepsilon_M)$ on which a semiring K acts from the left such that

$$(c \cdot t) \circ \langle t_1, \dots, t_{\mathrm{rk}_M(t)} \rangle = c \cdot (t \circ \langle t_1, \dots, t_{\mathrm{rk}_M(t)} \rangle).$$

If we remove ε from $(\Sigma, \mathrm{rk})^*$ then it is easy to see that $(\mathrm{T}_{\Sigma'} \setminus \{\varepsilon\}, \mathrm{rk}^*, \circ)$ is still a ranked semigroup. We will denote it by $(\Sigma, \mathrm{rk})^+$.

If (Σ_A, rk) is a monadic ranked alphabet for the alphabet A, then we readily observe that $(\Sigma_A, \mathrm{rk})^*$ is essentially a usual monoid that is isomorphic to A^*—the free monoid generated by A, and that $(\Sigma, \mathrm{rk})^+$ is essentially a semigroup that is isomorphic to A^+—the free semigroup generated by A. Observe however, that $(\Sigma, \mathrm{rk})^+$ is in general not a free ranked semigroup.

2 Weighted Tree-Languages

In this section we introduce the central concept of this thesis—weighted treelanguages. After the main definitions we recall some basic knowledge from category theory that is useful in our context. Following this we have a look onto limits and colimits in the category of weighted treelanguages.

The biggest part of the section is occupied by the introduction of operations on weighted tree-languages such as topcatenation, a-product, a-iteration etc. Each operation is modeled by a functor on the category of weighted tree-languages and we show that each of our functors is well-behaved in that it preserves directed (if not arbitrary) colimits. Since the categorial definitions of our operations are sometimes a bit opaque, we often offer another, more transparent, construction and prove its adequacy.

At the end of the section we introduce the two important concepts of finitarity and quasiregularity for weighted treelanguages and show some of their connections. Later, when we prove a connection between weighted treelanguages and formal tree-series, these terms will be essential.

In the sequel let Σ always denote a ranked alphabet and $K = (K, \oplus, \odot, 0, 1)$ be a semiring.

2.1 Weighted tree-languages. A *weighted (Σ-) tree-language* is a pair $(L, |.|)$ where L is a set and $|.| : L \longrightarrow \mathrm{WT}_\Sigma$, $s \mapsto |s|$. Let $\mathcal{L}_1 = (L_1, |.|_1)$ and $\mathcal{L}_2 = (L_2, |.|_2)$ be weighted tree-languages. A function $h : L_1 \longrightarrow L_2$ is called *homomorphism* from $\mathcal{L}_1$ to $\mathcal{L}_2$ if the following diagram commutes:

$$\begin{tikzcd} L_1 \arrow[rr, "h"] \arrow[dr, "|.|_1"'] & & L_2 \arrow[dl, "|.|_2"] \\ & \mathrm{WT}_\Sigma & \end{tikzcd}$$

With this definition the weighted tree-languages form a category. It will be denoted by WTL_Σ.

2.2 Some categorial notions. Before studying WTL_Σ we recall some notions from category-theory. Let C be a category. If $X, Y, Z \in \mathsf{C}$ are objects and if $f : X \longrightarrow Y$, $g : Y \longrightarrow Z$ are morphisms, then we denote the composition of f and g by $g \circ f : X \longrightarrow Z$. The identity-morphism of an object X will be denoted by $\mathbf{1}_X$. A morphism $f : X \longrightarrow Y$ is called *monomorphism* (or short: *mono*) if for all $U \in \mathsf{C}$ and for all $g, h : U \longrightarrow X$ holds $f \circ g = f \circ h \Rightarrow g = h$. A mono $f : X \longrightarrow Y$ is called split-mono if there is a morphism $f' : Y \longrightarrow X$ such that $f' \circ f = \mathbf{1}_X$. Dually, a morphism $f : X \longrightarrow Y$ is called *epimorphism* (or short: *epi*), if for all $U \in \mathsf{C}$, $g, h : Y \longrightarrow U$ holds $g \circ f = h \circ f \Rightarrow g = h$. An epi $f : X \longrightarrow Y$ is called *split-epi* if there exists a morphism $f' : Y \longrightarrow X$ such that $f \circ f' = \mathbf{1}_Y$. A morphism $f : X \longrightarrow Y$ is called *isomorphism* if there is a morphism $f' : Y \longrightarrow X$ such that $f \circ f' = \mathbf{1}_Y$ and $f' \circ f = \mathbf{1}_X$. With $\mathsf{C}(X, Y)$ we will denote the collection of all morphisms from X to Y (such collection of morphisms are sometimes called *hom-sets*).

A category C is called *locally small* if each hom-set is a set. It is called *small* if it is locally small and if its objects form a set.

A (small) *diagram* is a functor D from some (small) *category* I to C. A diagram $D : \mathsf{I} \longrightarrow \mathsf{C}$ is called *upwards directed* if I is an upward directed partially ordered set (poset)—that is in I any two elements have an upper bound. If I is a chain that is isomorphic to the least infinite ordinal ω, then D is called ω-*cochain* (or *sequence*). An *omega*-cochain is well-defined by the family $(Di, D(i, i{+}1))_{i \in \omega}$ where $(i, i + 1)$ denotes the morphism in ω that corresponds to prime-interval from i to its successor $i + 1$ in ω. Since $(\mathbb{N}, \leq)$ is isomorphic to ω, we shall sometimes take the freedom of identifying the two well-orderings.

A diagram D is called *downwards directed* if I is a downwards directed poset—that is, any two elements of I have a lower bound.

A diagram $D : I \longrightarrow \mathsf{C}$ is called *discrete* if I has no morphisms except the units.

A *compatible cocone* (or for short *cocone*) for a diagram D is a pair $(A, (f_i)_{i \in \mathsf{I}})$ such that $A \in \mathsf{C}$, $f_i : Di \longrightarrow A$ $(i \in \mathsf{I})$ and for each I-morphism $d : i \longrightarrow j$ the following diagram commutes:

$$\begin{tikzcd} & A & \\ Di \arrow[ur, "f_i"] \arrow[rr, "Dd"'] & & Dj \arrow[ul, "f_j"'] \end{tikzcd}$$

It is called *limiting cocone* of D if for any other cocone $(B, (g_i)_{i \in \mathsf{I}})$ of D there exists a unique arrow $! : A \longrightarrow B$ such that for all $i \in \mathsf{I}$ the following diagram commutes:

$$\begin{tikzcd} A \arrow[rr, dashed, "!"] & & B \\ & Di \arrow[ul, "f_i"] \arrow[ur, "g_i"'] & \end{tikzcd}$$

In this case the object A is called *colimit* of D (denoted by $\operatorname{colim} D$). Colimits of upward directed diagrams are called *directed colimits*. Those of discrete diagrams are called *coproducts*. Colimits of discrete diagrams are usually denoted by

$$\coprod_{i \in \mathsf{I}} Di.$$

Colimits are unique up to isomorphism. A category is called *cocomplete* if each small diagram has a colimit. It is called ω-*cocomplete* if each ω-cochain has a colimit.

A functor $F : \mathsf{C} \longrightarrow \mathsf{D}$ *preserves* the colimit of the diagram D if for each limiting cocone $(A, (f_i)_{i \in \mathsf{I}})$ the cocone $(FA, (Ff_i)_{i \in \mathsf{I}})$ is a limiting for $F \circ D$. We say that F

preserves colimits if it preserves colimits of all small diagrams. If F preserves the colimits of all ω-cochains, then it is called ω^{op}*-continuous*.

We say that F *creates* colimits if for each colimit $(B, (g_i)_{i \in \mathsf{I}})$ of $F \circ D$ there is a unique cocone $(A, (f_i)_{i \in \mathsf{I}})$ of D such that $(FA, (Ff_i)_{i \in \mathsf{I}}) = (B, (g_i)_{i \in \mathsf{I}})$, and moreover, $(A, (f_i)_{i \in \mathsf{I}})$ is a colimit of D. It is immediately clear that, if D is cocomplete and F creates colimits, then C is cocomplete as well. A functor $G : \mathsf{C} \longrightarrow \mathsf{C}$ is called *lifting* of a functor $H : \mathsf{D} \longrightarrow \mathsf{D}$ along F if the following diagram commutes:

$$\begin{tikzcd} \mathsf{C} \arrow[r, "G"] \arrow[d, "F"'] & \mathsf{C} \arrow[d, "F"] \\ \mathsf{D} \arrow[r, "H"'] & \mathsf{D} \end{tikzcd}$$

In this situation suppose that F creates colimits, D is cocomplete and H preserves colimits. Then G preserves colimits as well.

Limits of diagrams are defined dually to colimits—by cones. A *compatible cone* (or short: *cone*) of a diagram $D : \mathsf{I} \longrightarrow \mathsf{C}$ is a pair $(A, (f_i)_{i \in \mathsf{I}})$ such that $A \in \mathsf{C}$, $f_i : A \longrightarrow Di$ $(i \in \mathsf{I})$ and for each I-morphism $d : i \longrightarrow j$ the following diagram commutes:

$$\begin{tikzcd} Di \arrow[rr, "Dd"] & & Dj \\ & A \arrow[ul, "f_i"] \arrow[ur, "f_j"'] & \end{tikzcd}$$

A cone is called *limiting cone* for D if for every other compatible cone $(B, (g_i)_{i \in \mathsf{I}})$ of D there is a unique morphism $! : B \longrightarrow A$ such that the following diagram commutes for all $i \in \mathsf{I}$:

$$\begin{tikzcd} & Di & \\ A \arrow[ur, "f_i"] & & B \arrow[ul, "g_i"'] \arrow[ll, dashed, "!"'] \end{tikzcd}$$

In this case B is called *limit* of D (denoted by $\lim D$).

An object I of C is called *initial object* if $\mathsf{C}(I, A)$ is a singleton for every $A \in \mathsf{C}$. An object T of C is called *terminal object* if $\mathsf{C}(A, T)$ is a singleton for every $a \in \mathsf{C}$.

We say that $F : \mathsf{C} \longrightarrow \mathsf{C}$ *preserves monos* if F maps monomorphisms to monomorphisms.

For $F : \mathsf{C} \longrightarrow \mathsf{C}$ an F*-algebra* is a pair (A, α) where $A \in \mathsf{C}$ is the carrier and where $\alpha : FA \longrightarrow A$. Dually, an F*-coalgebra* is a pair (B, β) where $B \in \mathsf{C}$ is the carrier and where $\beta : B \longrightarrow FB$. Both, the F-algebras and the F-coalgebras form categories. There, for two F-algebras (A_1, α_1) and (A_2, α_2) a morphism $f : A_1 \longrightarrow A_2$ is called homomorphism if $f \circ \alpha_1 = \alpha_2 \circ Ff$. Analogously, for two F-coalgebras (B_1, β_1) and (B_2, β_2) a morphism $f : B_1 \longrightarrow B_2$ is called homomorphism if $Ff \circ \beta_1 = \beta_2 \circ f$. Initial objects in the category of F-algebras will be called *initial F-algebras* (we may also use the classical term *free F-algebras* instead). The terminal objects in the category of F-coalgebras will be called *terminal F-coalgebras*.

2.3 Remark. The empty set equipped with the empty mapping to WT_Σ is the initial object in WTL_Σ. Moreover $(\mathrm{WT}_\Sigma, \mathbf{1}_{\mathrm{WT}_\Sigma})$ is a terminal object in WT_Σ.

Note also, that the monomorphisms of WTL_Σ are just the injective homomorphisms and that the epis in WTL_Σ are just the surjective homomorphisms. Moreover, all epis in WTL_Σ are split-epis. However, except for the isos no mono splits.

2.4 Lemma. *WTL_Σ has arbitrary colimits. The forgetful functor*

$$U : \mathsf{WTL}_\Sigma \longrightarrow \mathsf{Set} : (L, |.|) \mapsto L, \quad f \mapsto f$$

creates colimits. In particular the colimits in WTL_Σ may be constructed like in Set.

Proof. WTL_Σ is the category of (although trivial) coalgebras for the constant signature-functor C_{WT_Σ} (see [32, 29, 46] for the basics about coalgebras). Using a result by Rutten in [46] we observe that the forgetful functor creates colimits and hence WTL_Σ is cocomplete. $\square$

2.5 Limits in $\mathbf{WTL}_\Sigma$. Note that WTL_Σ is not only cocomplete but also complete. We could also argue with the general theory of coalgebras in order to come to this conclusion (cf. [39]). However, since our signature functor is just a constant functor, we are not going to use this heavy machinery. Instead we give a self-contained description of the construction of limits of weighted tree-languages.

Let $D : \mathsf{I} \longrightarrow \mathsf{WTL}_\Sigma$ be a diagram. For $i \in \mathsf{I}$ let us denote the language Di by $(L_i, |.|_i)$. Let us define a set L consisting of all families $(t_i)_{i \in \mathsf{I}}$ such that

1. $\forall i : t_i \in L_i$,
2. $\forall i, j : |t_i|_i = |t_j|_j$,
3. $\forall i, j\, \forall f \in \mathsf{I}(i, j) : (Df)(t_i) = t_j$.

Because of 2 it makes sense to define $|(t_i)_{i \in I}| := |t_i|_i$ for some $i \in I$. Now $(L, |.|) = \lim D$, but since limits are not going to play an essential role in the sequel, we skip the proof of this claim. In order to make this construction a little bit better understandable let us finally have a look onto special limits—the products. Let $\mathcal{L}_1 = (L_1, |.|_1)$, $\mathcal{L}_2 = (L_2, |.|_2)$. Define

$$L := \{(t_1, t_2) \in L_1 \times L_2 \mid |t_1|_1 = |t_2|_2\}$$

and set $|(t_1, t_2)| := |t_1|_1 = |t_2|_2$. Then $(L, |.|) = \mathcal{L}_1 \times \mathcal{L}_2$. Note that the product plays the role of a generalized intersection of weighted tree-languages just like the coproduct is a generalization of the union-operation[3].

[3] In fact the product is also closely related to the Hadamard-product on formal tree-series

2.6 Product with scalars on WT_Σ. The action of K on WT_Σ may be extended to an action of K on WTL_Σ according to $c \cdot (L, |.|) := (L, |.|')$ where $|.|' : t \mapsto c \cdot |t|$.

2.7 Proposition. *Multiplying with scalars is functorial. The functor $[c \cdot -] : \mathcal{L} \mapsto c \cdot \mathcal{L}$, $h \mapsto h$ preserves arbitrary colimits and monos.*

Proof. Functoriality is obvious. Let $U : \mathsf{WTL}_\Sigma \longrightarrow \mathsf{Set}$ be the forgetful functor. Then $[c \cdot -]$ is a lifting of the identical functor on Set along U. Hence it preserves arbitrary colimits and monos. $\square$

2.8 Topcatenation. Next we define *topcatenation* of weighted tree-languages. Given $f \in \Sigma^{(n)}$, $c \in K$ and $\mathcal{L}_1 = (L_1, |.|_1), \ldots, \mathcal{L}_n = (L_n, |.|_n)$ weighted tree languages. Then $[f|c]\langle \mathcal{L}_1, \ldots, \mathcal{L}_n \rangle := (L, |.|_{[f|c]})$ where $L = L_1 \times \cdots \times L_n$ and $|(t_1, \ldots, t_n)|_{[f|c]} := [f|c]\langle |t_1|_1, \ldots, |t_n|_n \rangle$.

2.9 Proposition. *For a fixed $f \in \Sigma^{(n)}$ and $c \in K$ topcatenation with $[f|c]$ is an n-ary multifunctor of WTL_Σ into itself. Moreover, this functor $[f|c]\langle -_1, \ldots, -_n \rangle$ preserves arbitrary colimits and monos in each coordinate.*

Proof. On morphisms $[f|c]\langle -_1, \ldots, -_n \rangle$ acts by mapping $(h_1, \ldots, h_n)$ to $h_1 \times \cdots \times h_n$.

It is sufficient to check functoriality in one coordinate. Then, by analogy, it is functorial in each coordinate and hence it is functorial altogether.

Let us only consider the first coordinate: Suppose $\mathcal{L}_1, \ldots, \mathcal{L}_n$ and $\mathcal{L}_1'$ are weighted tree-languages and $h : \mathcal{L}_1 \longrightarrow \mathcal{L}_1'$ is a homomorphism. We show that $h \times \mathbf{1}_{\mathcal{L}_2} \times \cdots \times \mathbf{1}_{\mathcal{L}_n}$ is a homomorphism from $[f|c]\langle \mathcal{L}_1, \ldots, \mathcal{L}_n \rangle$ to $[f|c]\langle \mathcal{L}_1', \mathcal{L}_2, \ldots, \mathcal{L}_n \rangle$:

$$\begin{aligned} |(h \times \mathbf{1}_{\mathcal{L}_2} \times \cdots \times \mathbf{1}_{\mathcal{L}_n})(t_1, \ldots, t_n)| &= |(h(t_1), t_2, \ldots, t_n)| \\ &= [f|c]\langle |h(t_1)|_1, \ldots, |t_n|_n \rangle \\ &= [f|c]\langle |t_1|_1, \ldots, |t_n|_n \rangle. \end{aligned}$$

Compatibility with composition follows from the fact that the direct product is functorial.

It remains to show that topcatenation preserves colimits and monos in all variables. For $F = [f|c]\langle -, \mathcal{L}_2, \ldots, \mathcal{L}_n \rangle$ this follows from the fact that F is a lifting of the functor $X \mapsto X \times (L_2 \times \cdots \times L_n)$ which preserves colimits and monos. $\square$

2.10 a-Product of weighted trees with weighted tree-languages. Let $a \in \Sigma^{(0)}$. Next we define an action of WT_Σ on WTL_Σ from the left—the *a-Product*. Intuitively $t \cdot_a \mathcal{L}$ is obtained from $\mathcal{L}$ by constructing all possible weighted trees $t \circ_a \langle t_1, \ldots, t_n \rangle$ for $t_1, \ldots, t_n \in \mathcal{L}$ (respecting multiplicities in the multiset $\mathcal{L}$). The exact definition proceeds by induction on the structure of t: $[a|c] \cdot_a \mathcal{L} := c \cdot \mathcal{L}$, $[b|c] \cdot_a \mathcal{L} := \{[b|c]\}$ and $[f|c]\langle t_1, \ldots, t_n \rangle \cdot_a \mathcal{L} := [f|c]\langle t_1 \cdot_a \mathcal{L}, \ldots, t_n \cdot_a \mathcal{L} \rangle$

Let $\mathcal{L}_1 = (L_1, |.|_1)$ and $\mathcal{L}_2 = (L_2, |.|_2)$ be weighted tree-languages. Let $t \in \mathrm{WT}_\Sigma$ and let $h : \mathcal{L}_1 \longrightarrow \mathcal{L}_2$ be a homomorphism. Then we define $t \cdot_a h : t \cdot_a \mathcal{L}_1 \longrightarrow t \cdot_a \mathcal{L}_2$

by induction on t: $[a|c] \cdot_a h : [a|c] \cdot_a \mathcal{L}_1 = c \cdot \mathcal{L}_1 \longrightarrow c \cdot \mathcal{L}_2 = [a|c] \circ \mathcal{L}_2$ is defined by $c \cdot t \mapsto c \cdot h(t)$. $[b|c] \cdot_a h := \mathrm{id}_{\{[b|c]\}}$. And $[f|c]\langle t_1, \dots, t_n\rangle \cdot_a h := [f|c]\langle t_1 \cdot_a h, \dots, t_n \cdot_a h\rangle$.

2.11 Proposition. *For a given $t \in \mathrm{WT}_\Sigma$ and $a \in \Sigma^{(0)}$ the assignments $\mathcal{L} \mapsto t \cdot_a \mathcal{L}$, $h \mapsto t \cdot_a h$ describe an endofunctor of WTL_Σ that preserves directed colimits and monos.*

Proof. We proceed by induction on the structure of t.

For $[a|c] \cdot_a -$ this follows from 2.7.

$[b|c] \cdot_a -$ is a constant functor that maps every weighted tree-language to $\{[b|c]\}$ and that maps every morphism to $\mathbf{1}_{\{[b|c]\}}$. But such functors obviously preserve directed colimits and monos.

For the remaining case we argue as follows:

$$\begin{aligned}
[f|c]\langle t_1, \dots, t_n\rangle \cdot_a (h \circ g) &= [f|c]\langle t_1 \cdot_a (h \circ g), \dots, t_n \cdot_a (h \circ g)\rangle \\
&= [f|c]\langle (t_1 \cdot_a h) \circ (t_1 \cdot_a g), \dots, (t_n \cdot_a h) \circ (t_n \cdot_a g)\rangle \\
&= [f|c]\langle t_1 \cdot_a h, \dots, t_n \cdot_a h\rangle \circ [f|c]\langle t_1 \cdot_a g, \dots, t_n \cdot_a g\rangle \\
&= [f|c]\langle t_1, \dots, t_n\rangle \cdot_a h \circ [f|c]\langle t_1, \dots, t_n\rangle \cdot_a g.
\end{aligned}$$

Topcatenation with $[f|c]$ preserves arbitrary colimits and monos. By induction hypothesis each of the functors $[t_1 \cdot_a -], \dots, [t_n \cdot_a -]$ preserves directed colimits and monos. Hence the tupling $\langle [t_1 \cdot_a -], \dots, [t_n \cdot_a -]\rangle$ preserves directed colimits and monos. Consequently

$$\begin{aligned}
[[f|c]\langle t_1, \dots, t_n\rangle \cdot_a -] &= [f|c]\langle [t_1 \cdot_a -], \dots, [t_n \cdot_a -]\rangle \\
&= [f|c]\langle -_1, \dots, -_n\rangle \circ \langle [t_1 \cdot_a -], \dots, [t_n \cdot_a -]\rangle
\end{aligned}$$

preserves directed colimits and monos (note that functors preserving directed colimits and monos are closed under composition). □

2.12 Lemma. *Let $t \in \mathrm{WT}_\Sigma$ with $\mathrm{rk}_a(t) = n$. Let $\mathcal{L} = (L, |.|) \in \mathsf{WTL}_\Sigma$. Then*

$$t \cdot_1 \mathcal{L} \cong (L^n, |.|_{t,a})$$

where $|(s_1, \dots, s_n)|_{t,a} := t \circ_a \langle |s_1|, \dots, |s_n|\rangle$.

Proof. We proceed by induction on the structure of t. If $t = [a|c]$, then $t \cdot_a \mathcal{L} = c \cdot \mathcal{L} = (L, |.|')$ where $|s|' = c \cdot |s|$. On the other hand $|s|_{t,a} = t \circ_a \langle |s|\rangle = c \cdot |s|$. Hence $(L, |.|_{t,a}) \cong t \cdot_a \mathcal{L}$.

If $t = [b|c]$ then $\mathrm{rk}_a(t) = 0$ and $t \cdot_a \mathcal{L} = \{[b|c]\}$ On the other hand $(L^0, |.|_{t,a}) = (\{*\}, |.|_{t,a})$ with $|*|_{t,a} = t \circ_a \langle\rangle = [b|c] \circ_a \langle\rangle = [b|c]$. Hence $(L^0, |.|_{t,a}) \cong t \cdot_a \mathcal{L}$.

Assume now that $t = [f|c]\langle t_a, \dots, t_k\rangle$ where $\mathrm{rk}_a(t_i) = n_i$ $(1 \le i \le k)$ and $\mathrm{rk}_a(t) = \sum_{i=1}^n n_i =: n$. Then $L^n \cong L^{n_1} \times \dots \times L^{n_k}$ and

$$\begin{aligned}
|(s_{1,1} \dots, s_{1,n_1}, \dots, s_{k,n_k})|_{t,a} &= t \circ_a \langle |s_{1,1}|, \dots, |s_{k,n_k}|\rangle \\
&= [f|c]\langle t_1, \dots, t_k\rangle \circ_a \langle |s_{1,1}|, \dots, |s_{k,n_k}|\rangle \\
&= [f|c]\langle t_1 \circ_a \langle |s_{1,1}|, \dots, |s_{1,n_1}|\rangle, \dots, t_k \circ_a \langle |s_{k,1}|, \dots, |s_{k,n_k}|\rangle\rangle \\
&= [f|c]\langle |(s_{1,1}, \dots, s_{1,n_1})|_{t_1,a}, \dots, |(s_{k,1}, \dots, s_{k,n_k})|_{t_k,a}\rangle.
\end{aligned}$$

On the other hand, if $t_i \cdot_a L$ denotes the carrier of $t_i \cdot_a \mathcal{L}$ for $i = 1, \ldots, k$, then $[f|c]\langle t_1 \cdot_a \mathcal{L}, \ldots, t_k \cdot_a \mathcal{L}\rangle = (t_1 \cdot_a L \times \cdots \times t_k \cdot_a L, |.|_{[f|c]})$ where $|(r_1, \ldots, r_k)|_{[f|c]} = [f|c]\langle |r_1|, \ldots, |r_k|\rangle$. By induction hypothesis, $t_i \cdot_a \mathcal{L} \cong (L^{n_i}, |.|_{t_i,a})$. Hence $r_i = (s_{i,1}, \ldots, s_{i,n_i})$ for certain $s_{i,j} \in L$ and

$$[f|c]\langle |r_1|, \ldots, |r_k|\rangle = [f|c]\langle |(s_{1,1}, \ldots, s_{1,n_1})|_{t_1,a}, \ldots, |(s_{k,1}, \ldots, s_{k,n_k})|_{t_k,a}\rangle.$$

Hence the carrier of $t \cdot_a \mathcal{L}$ is L^n and $|.|_{[f|c]} = |.|_{t,a}$. Therefore $(L^n, |.|_{t,a}) \cong t \cdot_a \mathcal{L}$ □

2.13 General substitutions. Let $\varphi : \mathrm{WT}_\Sigma \longrightarrow \mathsf{WTL}_\Sigma$ be a function. Define $F_\varphi : \mathsf{WTL}_\Sigma \longrightarrow \mathsf{WTL}_\Sigma$ according to

$$\mathcal{L} \mapsto \coprod_{t \in \mathcal{L}} \varphi(|t|).$$

Let $\iota_t : \varphi(|t|) \rightarrowtail F_\varphi(\mathcal{L})$ be the coproduct-injections ($t \in \mathcal{L}$). Let $\mathcal{L}, \mathcal{L}' \in \mathsf{WTL}_\Sigma$ and let $h : \mathcal{L} \longrightarrow \mathcal{L}'$ be a homomorphism. For $t' \in \mathcal{L}'$ let $\iota'_{t'}$ be the coproduct-injection of $\varphi(|t'|)$ into $F_\varphi(\mathcal{L}')$. Since for all $t \in \mathcal{L}$ we have $\varphi(|t|) = \varphi(|h(t)|)$, we may define $F_\varphi(h)$ pointwise: In particular for $x \in \varphi(|t|)$ we define

$$F_\varphi(h)(\iota_t(x)) := \iota'_{h(t)}(x).$$

2.14 Proposition. *With the notions from above F_φ is a functor that preserves arbitrary colimits and monos.*

Proof. **functoriality:** Let $\mathcal{L}, \mathcal{L}', \mathcal{L}'' \in \mathsf{WTL}_\Sigma$ and let $h : \mathcal{L} \longrightarrow \mathcal{L}'$ and $g : \mathcal{L}' \longrightarrow \mathcal{L}''$ be homomorphisms. Moreover, let

$$\begin{aligned} \iota_t &: \varphi(|t|) \rightarrowtail F_\varphi(\mathcal{L}), \\ \iota'_t &: \varphi(|t|') \rightarrowtail F_\varphi(\mathcal{L}'), \\ \iota''_t &: \varphi(|t|'') \rightarrowtail F_\varphi(\mathcal{L}'') \end{aligned}$$

be the respective embeddings into the coproducts. Then for $x \in \varphi(|t|)$ we compute:

$$\begin{aligned} F_\varphi(g) \circ F_\varphi(h)(\iota_t(x)) &= F_\varphi(g)(\iota'_{h(t)}(x)) \\ &= \iota''_{g(h(t))}(x) \\ &= \iota''_{(g\circ h)(t)}(x) = F_\varphi(g \circ h)(\iota_t(x)). \end{aligned}$$

The fact that identities are preserved follows directly from the definition.

preservation of monos: Let $\mathcal{X}, \mathcal{Y} \in \mathsf{WTL}_\Sigma$, $f : \mathcal{X} \longrightarrow \mathcal{Y}$ be a mono. For $t \in \mathcal{X}$ let $\iota_t : \varphi(|t|) \longrightarrow F_\varphi(\mathcal{X})$ be the coproduct-embedding and for $s \in \mathcal{Y}$ let $\iota'_s : \varphi(|s|) \longrightarrow F_\varphi(\mathcal{Y})$ be the coproduct-embedding. Since a morphism in WTL_Σ with non empty domain is mono if and only if it is injective, we may distinguish two cases. The case where $\mathcal{X} = \emptyset$ implies that f is the empty mapping. But by construction F_φ preserves the empty set and thus it preserves also the empty function. So monos with empty domain are mapped to monos with empty domain. Assume now that $\mathcal{X} \neq \emptyset$. Then $F_\varphi(\mathcal{X}) \neq \emptyset$. Hence we need to show that $F_\varphi(f)$ is injective. Let $u, v \in F_\varphi(\mathcal{X})$. Then there exist unique $t \in \mathcal{X}$, $\bar{u} \in \varphi(|t|)$ with $u = \iota_t(\bar{u})$ and there exist unique $s \in \mathcal{X}$, $\bar{v} \in \varphi(|s|)$ with $v = \iota_s(\bar{v})$. Hence

$$F_\varphi(f)(u) = F_\varphi(f)(\iota_t(\bar{u})) = \iota'_{f(t)}(\bar{u})$$
$$F_\varphi(f)(v) = F_\varphi(f)(\iota_s(\bar{v})) = \iota'_{f(s)}(\bar{v}).$$

Since $\iota_{f(t)}$, $\iota_{f(s)}$ are coproduct-injections their images are disjoint if and only if $f(t) \neq f(s)$. Since f is injective, we derive

$$F_\varphi(f)(u) = F_\varphi(f)(v) \iff \iota_{f(t)}(\bar{u}) = \iota'_{f(s)}(\bar{v}) \Rightarrow f(t) = f(s) \Rightarrow t = s.$$

Since $\iota'_{f(t)}$ is injective, this implies $\bar{u} = \bar{v}$ and hence $u = v$. Hence $F_\varphi(f)$ is injective.

preservation of coproducts: Let I be a small category whose only morphisms are the identities and let $D : \mathsf{I} \longrightarrow \mathsf{WTL}_\Sigma : i \mapsto \mathcal{L}_i$ be a diagram. Suppose $(\mathcal{X}, (e_i)_{i\in\mathsf{I}})$ is a limiting cocone of D. Then we have to show that $(F_\varphi(\mathcal{X}), (F_\varphi(e_i))_{i\in\mathsf{I}})$ is a limiting cocone of $F_\varphi \circ D$. Let $(\mathcal{Y}, (f_i)_{i\in\mathsf{I}})$ be a limiting cocone of $F_\varphi \circ D$. We will show that the cocones $(F_\varphi(\mathcal{X}), (F_\varphi(e_i))_{i\in\mathsf{I}})$ and $(\mathcal{Y}, (f_i)_{i\in\mathsf{I}})$ are isomorphic. That is, we show that there is an isomorphism $g : \mathcal{Y} \longrightarrow F_\varphi(\mathcal{X})$ such that the following diagram commutes for all $i \in \mathsf{I}$:

$$\begin{tikzcd} F_\varphi(\mathcal{X}) \arrow[rr, shift left, "g^{-1}"] & & \mathcal{Y} \arrow[ll, shift left, "g"] \\ & F_\varphi(\mathcal{L}_i) \arrow[ul, "F_\varphi(e_i)"] \arrow[ur, "f_i"'] & \end{tikzcd} \tag{1}$$

A direct consequence of this would be that both cocones are limiting.

First we note that since $(\mathcal{Y}, (f_i)_{i\in\mathsf{I}})$ is a limiting cocone, there exists a unique $g : \mathcal{Y} \longrightarrow F_\varphi(\mathcal{X})$ such that the following diagram commutes:

$$\begin{tikzcd} F_\varphi(\mathcal{X}) & & \mathcal{Y} \arrow[ll, "g"'] \\ & F_\varphi(\mathcal{L}_i) \arrow[ul, "F_\varphi(e_i)"] \arrow[ur, "f_i"'] & \end{tikzcd} \tag{2}$$

Note that f_i is mono since it is a coproduct embedding and that $F_\varphi(e_i)$ is mono since e_i is mono and since F_φ preserves monos.

For $i \in \mathsf{I}$, $t \in \mathcal{L}_i$ let $\iota_{i,t} : \varphi(|t|) \rightarrowtail F_\varphi(\mathcal{L}_i)$ be the coproduct embedding and for $t \in \mathcal{X}$ let $\iota_t : \varphi(|t|) \rightarrowtail F_\varphi(\mathcal{X})$ be the coproduct embedding. Since $\mathcal{X}$ is a colimit of D, for each $t \in \mathcal{X}$ there exist unique $i \in I$, $t' \in \mathcal{L}_i$ such that $\iota_t = \iota_{e_i(t')}$. Hence for every $z \in F_\varphi(\mathcal{X})$ there exist unique $i \in \mathsf{I}$, $t \in \mathcal{L}_i$, $\bar{z} \in \varphi(|e_i(t)|)$ such that $z = \iota_{e_i(t)}(\bar{z})$. Since $\varphi(|e_i(t)|) = \varphi(|t|)$, we can take $\hat{z} := \iota_{i,t}(\bar{z})$ and define $h(z) := f_i(\hat{z})$. Since $|z| = |\bar{z}|$ and since f_i is a homomorphism, we have thus defined a homomorphism $h : F_\varphi(\mathcal{X}) \longrightarrow \mathcal{Y}$. Moreover the following diagram commutes for every $i \in \mathsf{I}$ by construction of h:

$$\begin{tikzcd} F_\varphi(\mathcal{X}) \arrow[rr, "h"] & & \mathcal{Y} \\ & F_\varphi(\mathcal{L}_i) \arrow[ul, "F_\varphi(e_i)"] \arrow[ur, "f_i"'] & \end{tikzcd} \tag{3}$$

Next we compute

$$\begin{aligned} (g \circ h)(z) &= g(f_i(\hat{z})) \\ &= F_\varphi(e_i)(\hat{z}) && \text{by (2)} \\ &= F_\varphi(e_i)(\iota_{i,t}(\bar{z})) \\ &= \iota_{e_i(t)}(\bar{z}) = z. \end{aligned}$$

Hence $g \circ h = \mathbf{1}_X$ and thus h is split-mono and g is split-epi.

On the other hand, by (3), the following diagram commutes for every $i \in \mathsf{I}$:

$$\begin{tikzcd} \mathcal{Y} \arrow[rr, "h \circ g"] & & \mathcal{Y} \\ & F_\varphi(\mathcal{L}_i) \arrow[ul, "f_i"] \arrow[ur, "f_i"'] & \end{tikzcd}$$

Hence, because $(\mathcal{Y}, (f_i)_{i \in \mathsf{I}})$ is a limiting cocone and since also

$$\begin{tikzcd} \mathcal{Y} \arrow[rr, "\mathbf{1}_\mathcal{Y}"] & & \mathcal{Y} \\ & F_\varphi(\mathcal{L}_i) \arrow[ul, "f_i"] \arrow[ur, "f_i"'] & \end{tikzcd}$$

commutes for all $i \in \mathsf{I}$, we conclude $h \circ g = \mathbf{1}_\mathcal{Y}$.

Hence h is split-epi and g is split-mono. Altogether g is an isomorphism and $h = g^{-1}$. Thus diagram (1) commutes.

preservation of coequalizers Let $\mathcal{X} = (X, |.|), \mathcal{Y} = (Y, |.|) \in \mathsf{WTL}_\Sigma$ and $f, g : \mathcal{X} \longrightarrow \mathcal{Y}$. By 2.4 the coequalizer of f and g may be computed like in Set. Let $\theta := \{(f(x), g(x)) \mid x \in X\}$, $\bar{\theta}$ be the smallest equivalence relation on Y containing θ. Then $Y/\bar{\theta}$ may be turned into a weighted tree-language by defining $|[t]_{\bar{\theta}}| := |t|$. Let us denote this language by $\mathcal{Y}/\bar{\theta}$. The structure map is well defined since for all $x \in X$ we have $|f(x)| = |g(x)| = |x|$. This property is inherited by $\bar{\theta}$ where we have $(t, t') \in \bar{\theta} \Rightarrow |t| = |t'|$. We may conclude that the canonical epimorphism $\mathrm{nat}_{\bar{\theta}} : Y \longrightarrow Y/\bar{\theta}$ is a homomorphism from $\mathcal{Y} \longrightarrow \mathcal{Y}/\bar{\theta}$. By 2.4 we have that $(\mathcal{Y}/\bar{\theta}, \mathrm{nat}_{\bar{\theta}} \circ f, \mathrm{nat}_{\bar{\theta}})$ is a coequalizer of f and g. Note that the apparent asymmetry in this construction between f and g is not real since we have $\mathrm{nat}_{\bar{\theta}} \circ f = \mathrm{nat}_{\bar{\theta}} \circ g$ by construction. Let us sum up the situation in the following commuting diagram:

$$\begin{tikzcd} & & \mathcal{Y}/\bar{\theta} \\ \mathcal{X} \arrow[rru, bend left, "\mathrm{nat}_{\bar{\theta}} \circ f"] \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & \mathcal{Y} \arrow[ur, "\mathrm{nat}_{\bar{\theta}}"'] & \end{tikzcd}$$

Obviously the diagram above still commutes after application of F_φ:

$$\begin{tikzcd}
 & & F_\varphi(\mathcal{Y}/\bar\theta) \\
F_\varphi(\mathcal{X}) \arrow[r, shift left, "F_\varphi(f)"] \arrow[r, shift right, "F_\varphi(g)"'] \arrow[urr, bend left, "F_\varphi(\mathrm{nat}_{\bar\theta}\circ f)"] & F_\varphi(\mathcal{Y}) \arrow[ur, "F_\varphi(\mathrm{nat}_{\bar\theta})"'] &
\end{tikzcd}$$

Now we need to show that $(F_\varphi(\mathcal{Y}/\bar\theta), F_\varphi(nat_{\bar\theta} \circ f), F_\varphi(\mathrm{nat}_{\bar\theta}))$ is again a coequalizer of $F_\varphi(f)$ and $F_\varphi(g)$.

By the same method as above we construct a coequalizer of $F_\varphi(f)$ and $F_\varphi(g)$. Let

$$\varrho := \{(F_\varphi(f)(t), F_\varphi(g)(t)) \mid t \in F_\varphi(\mathcal{X})\},$$

$\bar\varrho$ be the smallest equivalence relation containing ϱ and take its natural epimorphism $\mathrm{nat}_{\bar\varrho} : F_\varphi(\mathcal{Y}) \longrightarrow (F_\varphi(\mathcal{Y}))/\bar\varrho$. Then $(F_\varphi(\mathcal{Y})/\bar\varrho, \mathrm{nat}_{\bar\varrho} \circ F_\varphi(f), \mathrm{nat}_{\bar\varrho})$ is a coequalizer of $F_\varphi(f)$ and $F_\varphi(g)$. Thus there is a unique homomorphism $\xi : F_\varphi(\mathcal{Y})/\bar\varrho \longrightarrow F_\varphi(\mathcal{Y}/\bar\theta)$ such that the following diagram commutes:

$$\begin{tikzcd}
 & & F_\varphi(\mathcal{Y}/\bar\theta) \\
F_\varphi(\mathcal{X}) \arrow[r, shift left, "F_\varphi(f)"] \arrow[r, shift right, "F_\varphi(g)"'] \arrow[urr, bend left, "F_\varphi(\mathrm{nat}_{\bar\theta}\circ f)"] \arrow[drr, bend right, "\mathrm{nat}_{\bar\varrho}\circ F_\varphi(f)"'] & F_\varphi(\mathcal{Y}) \arrow[ur, "F_\varphi(\mathrm{nat}_{\bar\theta})"'] \arrow[dr, "\mathrm{nat}_{\bar\varrho}"] & \\
 & & F_\varphi(\mathcal{Y})/\bar\varrho \arrow[uu, "\xi"']
\end{tikzcd}$$

Next we construct a homomorphism $\psi : F_\varphi(\mathcal{Y}/\bar\theta) \longrightarrow F_\varphi(\mathcal{Y})/\bar\varrho$. We do this pointwise: Let $\iota'_{[y]_{\bar\theta}}$ be the embedding of $\varphi(|[y]_{\bar\theta}|)$ into $F_\varphi(\mathcal{Y}/\bar\theta)$ and let ι_y be the embedding of $\varphi(|y|)$ into $F_\varphi(\mathcal{Y})$. Let $s \in F_\varphi(\mathcal{Y}/\bar\theta)$. Without loss of generality assume $s = \iota'_{[y]_{\bar\theta}}(\hat s)$ for $\hat s \in \varphi(|[y]_{\bar\theta}|)$. Then we define $\psi(s) := [(\iota_y(\hat s))]_{\bar\varrho}$. We need to show that ψ is welldefined. In principle we must prove for this that $[y']_{\bar\theta} = [y]_{\bar\theta}$ implies $[\iota_y(\hat s)]_{\bar\varrho} = [\iota_{y'}(\hat s)]_{\bar\varrho}$. However, it is sufficient to show $(y, y') \in \theta \Rightarrow (\iota_y(\hat s), \iota_{y'}(\hat s)) \in \varrho$ because θ and ϱ generate $\bar\theta$ and $\bar\varrho$, respectively. Let $x \in \mathcal{X}$ such that $f(x) = y$ and $g(x) = y'$. By definition we have

$$\begin{aligned} \iota_{f(x)}(\hat s) &= F_\varphi(f)(\iota_x(\hat s)) \qquad \text{and} \\ \iota_{g(x)}(\hat s) &= F_\varphi(g)(\iota_x(\hat s)). \end{aligned}$$

Hence $(\iota_y(\hat s), \iota_{y'}(\hat s)) \in \varrho$ and consequently ψ is welldefined.

By construction of ψ the following diagram commutes:

$$\begin{tikzcd}
 & & F_\varphi(\mathcal{Y}/\bar\theta) \arrow[dd, "\psi"] \\
F_\varphi(\mathcal{X}) \arrow[r, shift left, "F_\varphi(f)"] \arrow[r, shift right, "F_\varphi(g)"'] \arrow[urr, bend left, "F_\varphi(\mathrm{nat}_{\bar\theta}\circ f)"] \arrow[drr, bend right, "\mathrm{nat}_{\bar\varrho}\circ F_\varphi(f)"'] & F_\varphi(\mathcal{Y}) \arrow[ur, "F_\varphi(\mathrm{nat}_{\bar\theta})"'] \arrow[dr, "\mathrm{nat}_{\bar\varrho}"] & \\
 & & F_\varphi(\mathcal{Y})/\bar\varrho
\end{tikzcd}$$

From $\psi \circ \xi \circ \mathrm{nat}_{\bar{\varrho}} = \psi \circ F_\varphi(\mathrm{nat}_{\bar{\theta}}) = \mathrm{nat}_{\bar{\varrho}}$ follows that $\psi \circ \xi = \mathbf{1}_{F_\varphi(\mathcal{Y})/\bar{\varrho}}$ because $F_\varphi(\mathcal{Y})/\bar{\varrho}$ is a colimit. Hence ξ is split-mono and ψ is split-epi. On the other hand it is easily seen that $F_\varphi(\mathrm{nat}_{\bar{\theta}})$ is split-epi because $\mathrm{nat}_{\bar{\theta}}$ is split-epi. Moreover $\mathrm{nat}_{\bar{\varrho}}$ is split-epi. Hence ξ is split-epi as well. Consequently ξ is an isomorphism. This completes the proof that F_φ preserves coequalizers. Hence, by a general result of category theory, F_φ preserves arbitrary colimits (cf. [7]). □

2.15 a-Product in WTL_Σ. The a-product may be extended to a binary operation on WTL_Σ with the methods from 2.13. In particular, for $\mathcal{L}_2 \in \mathsf{WTL}_\Sigma$, we may define $\varphi : \mathrm{WT}_\Sigma \longrightarrow \mathsf{WTL}_\Sigma$ according to

$$\varphi : t \mapsto t \cdot_a \mathcal{L}_2.$$

Then, by the previous lemma $F_\varphi : \mathcal{L}_1 \mapsto \mathcal{L}_1 \cdot_a \mathcal{L}_2 := \coprod_{t \in \mathcal{L}_1} \varphi(|t|)$ is a functor that preserves arbitrary colimits.

On the other hand, with 2.11, it is easy to see that

$$(\mathcal{L}_1 \cdot_a -) : \mathcal{L}_2 \mapsto \mathcal{L}_1 \cdot_a \mathcal{L}_2 = \coprod_{t \in \mathcal{L}_1} |t| \cdot_a \mathcal{L}_2$$

is a functor that preserves directed colimits (recall that functors preserving directed colimits are closed under coproducts). Summing up we showed that $(-_1 \cdot_a -_2) : \mathsf{WTL}_\Sigma^2 \longrightarrow \mathsf{WTL}_\Sigma$ is a functor that preserves arbitrary colimits in the first and directed colimits in the second coordinate.

2.16 Lemma. *Let $\mathcal{L}_1 = (L_1, |.|_1)$, $\mathcal{L}_2 = (L_2, |.|_2)$ be weighted tree-languages over Σ and let $a \in \Sigma^{(0)}$. Define a formal language L over the alphabet $L_1 \cup L_2 \cup \{,\} \cup \{<, >, \circ_a\}$ according to:*

$$L := \{t \circ_a \langle s_1, \dots, s_n\rangle \mid t \in L_1,\, s_1, \dots, s_n \in L_2,\, \mathrm{rk}_a(|t|) = n\}$$

and define $|.| : L \longrightarrow \mathrm{WT}_\Sigma$ by

$$|t \circ_a \langle s_1, \dots, s_n\rangle| := |t|_1 \circ_a \langle |s_1|_2, \dots, |s_n|_2\rangle.$$

Then $(L, |.|) \cong \mathcal{L}_1 \cdot_a \mathcal{L}_2$.

Proof. For $t \in L_1$ with $\mathrm{rk}_a(t) = n$ define $L_t \subseteq L$ by

$$L_t := \{t \circ_a \langle s_1, \dots, s_n\rangle \mid s_1, \dots, s_n \in L_2\}.$$

Then $(L_t, |.|_{|L_t})$ is a sublanguage of $(L, |.|)$. Observe that $L = \bigcup_{t \in L_1} L_t$ and that this union is disjoint. Hence $|.| = \bigcup_{t \in L_1} |.|_{|L_t}$. This shows that

$$(L, |.|) = \coprod_{t \in L_1} (L_t, |.|_{|L_t}).$$

Since obviously $(L_t, |.|_{|L_t}) \cong (L_2^n, |.|_{t,a}) \cong |t|_1 \cdot_a \mathcal{L}_2$ (cf. 2.12), we conclude that

$$(L, |.|) \cong \coprod_{t \in L_1} |t|_1 \cdot_a \mathcal{L}_2 = \mathcal{L}_1 \cdot_a \mathcal{L}_2.$$

□

2.17 Lemma. *Let $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3 \in \mathsf{WTL}_\Sigma$ and let $a \in \Sigma^{(0)}$. Then*

$$\mathcal{L}_1 \cdot_a (\mathcal{L}_2 \cdot_a \mathcal{L}_3) \cong (\mathcal{L}_1 \cdot_a \mathcal{L}_2) \cdot_a \mathcal{L}_3.$$

Proof.

$$\begin{aligned}
\mathcal{L}_1 \cdot_a (\mathcal{L}_2 \cdot_a \mathcal{L}_3) &= \coprod_{t \in \mathcal{L}_1} |t| \cdot_a (\mathcal{L}_2 \cdot_a \mathcal{L}_3) = \coprod_{t \in \mathcal{L}_1} |t| \cdot_a \left(\coprod_{s \in \mathcal{L}_2} |s| \cdot_a \mathcal{L}_2 \right) \\
&\cong \coprod_{t \in \mathcal{L}_1} |t| \cdot_a \left(\{ s \circ_a \langle b_1, \dots, b_{\mathrm{rk}_a(s)} \rangle \mid s \in \mathcal{L}_2,\, b_1, \dots, b_{\mathrm{rk}_a(s)} \in \mathcal{L}_3 \}, |.| \right) \\
&\cong \Big(\{ t \circ_a \langle s_1 \circ_a \langle b_{1,1}, \dots, b_{1,m_1} \rangle, \dots, s_n \circ_a \langle b_{n,1}, \dots, b_{n,m_n} \rangle \rangle \mid \\
&\qquad t \in \mathcal{L}_1,\, n = \mathrm{rk}_a(t),\, s_i \in \mathcal{L}_2,\, m_i = \mathrm{rk}_a(s_i),\, b_{i,j} \in \mathcal{L}_3, \\
&\qquad\qquad 1 \le i \le n,\, 1 \le j \le m_i \}, |.| \Big) \\
&\text{and by superassociativity of } \circ_a \\
&\cong \Big(\{ (t \circ_a \langle s_1, \dots, s_n \rangle) \circ_a \langle b_{1,1}, \dots, b_{n,m_n} \rangle \mid \dots \}, |.| \Big) \\
&\cong (\mathcal{L}_1 \cdot_a \mathcal{L}_2) \cdot_a \mathcal{L}_2
\end{aligned}$$

where

$$\begin{aligned}
&|t \circ_a \langle s_1 \circ_a \langle b_{1,1}, \dots, b_{1,m_1} \rangle, \dots, s_n \circ_a \langle b_{n,1}, \dots, b_{n,m_n} \rangle \rangle| \\
&\qquad := |t| \circ_a \langle |s_1| \circ_a \langle |b_{1,1}|, \dots, |b_{1,m_1}| \rangle, \dots, |s_n| \circ_a \langle |b_{n,1}|, \dots, |b_{n,m_n}| \rangle \rangle
\end{aligned}$$

and where

$$\begin{aligned}
&|(t \circ_a \langle s_1, \dots, s_n \rangle) \circ_a \langle b_{1,1}, \dots, b_{n,m_n} \rangle| \\
&\qquad := |t| \circ_a \langle |s_1|, \dots, |s_n| \rangle \circ_a \langle |b_{1,1}|, \dots, |b_{n,m_n}| \rangle.
\end{aligned}$$

□

2.18 Lemma. *Let $f \in \Sigma^{(n)}$, $c \in K$, $\mathcal{L}, \mathcal{L}_1, \dots, \mathcal{L}_n \in \mathsf{WTL}_\Sigma$ and $a \in \Sigma^{(0)}$. Then*

$$[f|c]\langle \mathcal{L}_1 \cdot_a \mathcal{L}, \dots, \mathcal{L}_n \cdot_a \mathcal{L} \rangle \cong [f|c]\langle \mathcal{L}_1, \dots, \mathcal{L}_n \rangle \cdot_a \mathcal{L}.$$

Proof.

$$\begin{aligned}
[f|c]\langle \mathcal{L}_1 \cdot_a \mathcal{L}, \dots, \mathcal{L}_n \cdot_a \mathcal{L}\rangle &= [f|c]\left\langle \coprod_{s_1\in\mathcal{L}_1} |s_1|_1 \cdot_a \mathcal{L}, \dots, \coprod_{s_n\in\mathcal{L}_n} |s_n|_n \cdot_a \mathcal{L}\right\rangle \\
&\cong \coprod_{s_1\in\mathcal{L}_1} \cdots \coprod_{s_n\in\mathcal{L}_n} [f|c]\langle |s_1|_1 \cdot_a \mathcal{L}, \dots, |s_n|_n \cdot_a \mathcal{L}\rangle \\
&= \coprod_{s_1\in\mathcal{L}_1} \cdots \coprod_{s_n\in\mathcal{L}_n} [f|c]\langle |s_1|_1, \dots, |s_n|_n\rangle \cdot_a \mathcal{L} \\
&\cong \coprod_{(s_1,\dots,s_n)\in[f|c]\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle} |(s_1,\dots,s_n)|_{[f|c]} \cdot_a \mathcal{L} \\
&= [f|c]\langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle \cdot_a \mathcal{L}.
\end{aligned}$$

□

2.19 a-Iteration. Now we have the tools to introduce the notion of a-iteration. For this we define the functor $\mathrm{S}_a : \mathsf{WTL}_\Sigma^2 \longrightarrow \mathsf{WTL}_\Sigma : (X,\mathcal{L}) \mapsto (\mathcal{L}\cdot_a X) + \{[a|1]\}$. From the propositions above it follows that this functor preserves directed colimits. In particular it is ω^{op}-continuous in both coordinates. Since WTL_Σ has an initial object (the empty language), it follows that an initial $\mathrm{S}_a(-,\mathcal{L})$-algebra $\mu X.\mathrm{S}_a(X,\mathcal{L})$ exists. It may be constructed by taking the colimit of the initial cochain:

$$\emptyset \xrightarrow{\ !\ } \mathrm{S}_a(\emptyset,\mathcal{L}) \xrightarrow{\mathrm{S}_a(!,\mathbf{1}_\mathcal{L})} \mathrm{S}_a^2(\emptyset,\mathcal{L}) \xrightarrow{\mathrm{S}_a^2(!,\mathbf{1}_\mathcal{L})} \cdots$$

The carrier of this initial algebra is denoted by $\mathcal{L}_a^*$ and is called the *a-iteration* of $\mathcal{L}$.

2.20 Remark. In the sequel we will compare several initial cochains in order to prove the isomorphism of their colimits. Two ω-cochains $(\mathcal{L}_i,\alpha_i)_{i\in\omega}$ and $(\mathcal{S}_i,\beta_i)_{i\in\omega}$ of weighted treelanguages (where $\alpha_i : \mathcal{L}_i \longrightarrow \mathcal{L}_{i+1}$ and $\beta_i : \mathcal{S}_i \longrightarrow \mathcal{S}_{i+1}$ $(i \in \mathbb{N})$) are *isomorphic* if there is a family of isomorphisms $(\mu_i)_{i\in\omega}$ where $\mu_i : \mathcal{L}_i \longrightarrow \mathcal{S}_i$ such that for all $i \in \omega$ the following diagram commutes:

$$\begin{tikzcd}
\mathcal{L}_i \arrow[r, "\alpha_i"] \arrow[d, "\mu_i"'] & \mathcal{L}_{i+1} \arrow[d, "\mu_{i+1}"] \\
\mathcal{S}_i \arrow[r, "\beta_i"] & \mathcal{S}_{i+1}
\end{tikzcd}$$

If two ω-cochains are isomorphic, then they also have isomorphic colimits. This observation will be our main tool when we will compare the initial cochains of several different functors. However, showing the isomorphism of ω-cochains is a rather tedious task. The following lemma shows that WTL_Σ is friendly and relieves us of some work, at least if the α_i and the β_i $(i \in \omega)$ are all injective. Such ω-cochains are called *injective*.

2.21 Lemma. *With the notions from above suppose that α_i, β_i are injective ($i \in \omega$). Then $(\mathcal{L}_i, \alpha_i)_{i\in\omega} \cong (\mathcal{S}_i, \beta_i)_{i\in\omega}$ if and only if $\mathcal{L}_i \cong \mathcal{S}_i$ for all $i \in \omega$.*

Proof. The direction from left to right is trivial.

Let $\nu_i : \mathcal{L}_i \longrightarrow \mathcal{S}_i$ be isomorphisms ($i \in \omega$). We will define new isomorphisms $\mu_i : \mathcal{L}_i \longrightarrow \mathcal{S}_i$ such that the above diagram commutes. To this end we proceed by induction on i and define $\mu_0 := \nu_0$. Suppose now that μ_i is already defined. Then $\beta_i \circ \mu_i(\mathcal{L}_i)$ and $\nu_{i+1} \circ \alpha_i(\mathcal{L}_i)$ are two isomorphic sublanguages of $\mathcal{S}_{i+1}$. We define a transposition $\pi : \mathcal{S}_{i+1} \longrightarrow \mathcal{S}_{i+1}$ by

$$x \mapsto \begin{cases} \nu_{i+1} \circ \alpha_i(t) & x = \beta_i \circ \mu_i(t) \text{ for some } t \in \mathcal{L}_i \\ \beta_i \circ \mu_i(t) & x = \nu_{i+1} \circ \alpha_i(t) \text{ for some } t \in \mathcal{L}_i \\ x & \text{otherwise.} \end{cases}$$

Then π is obviously an automorphism of $\mathcal{S}_{i+1}$. Moreover $\pi \circ \nu_{i+1} \circ \alpha_i = \beta_i \circ \mu_i$. Hence, if we define $\mu_{i+1} := \pi \circ \nu_{i+1}$, then the above diagram commutes and we are finished. □

2.22 a-Iteration (alternative construction). Given a weighted tree-language $\mathcal{L} = (L, |.|)$ we assign to each $t \in \mathcal{L}$ its rank by $\mathrm{rk}_a(t) := \mathrm{rk}_a(|t|)$. This makes (L, rk_a) a ranked set. Let $\mathcal{M} := (L_a^*, \mathrm{rk}_\varepsilon, \circ_\varepsilon, \varepsilon)$ be the free ranked monoid generated by (L, rk_a). Let $|.|_a^*$ be the initial homomorphism from $\mathcal{M}$ to $(\mathrm{WT}_\Sigma, \mathrm{rk}_a, \circ_a, [a|1])$.

2.23 Proposition. *The weighted tree-language $(L_a^*, |.|_a^*)$ is isomorphic to $\mathcal{L}_a^*$.*

Proof. According to 1.20, the free ranked monoid $\mathcal{M}$ generated by (L, rk_a) has as carrier the set of all trees over the ranked set $(L \cup \{\varepsilon\}, \mathrm{rk}'_a)$ where $\varepsilon \notin L$, $(\mathrm{rk}'_a)_{|L} = \mathrm{rk}$, $\mathrm{rk}'_a(\varepsilon) = 0$. This set of trees may also be obtained the union of the following increasing sequence of formal tree-languages:

$$\begin{aligned} M_0 &:= \emptyset, \\ M_{i+1} &:= \left(\bigcup_{n\in\mathbb{N}} \{ f\langle g_1, \dots, g_n\rangle \mid f \in L^{(n)}, g_1, \dots, g_n \in M_i \} \right) \cup \{\varepsilon\} \\ L_a^* &:= \bigcup_{n\in\mathbb{N}} M_n. \end{aligned}$$

Note that all summands of the union in the definition of M_{i+1} are mutually disjoint.

The claim that $M_i \subseteq M_{i+1}$ ($i \in \mathbb{N}$) is proved by induction on i. Obviously $M_0 \subseteq M_1$. Now suppose $M_{i-1} \subseteq M_i$. Let $g \in M_i$. Then either $g = \varepsilon$ or $g = f\langle g_1, \dots, g_n\rangle$ for some $n \in \mathbb{N}$, $f \in L^{(n)}$ and $g_1 \dots, g_n \in M_{i-1}$. In the first case $\varepsilon = g \in M_{i+1}$ by construction. In the latter case we argue that $g_1, \dots, g_n \in M_i$ by hypothesis. Hence $g = f\langle g_1, \dots, g_n\rangle \in M_{i+1}$ by construction.

Each M_i may be turned into a weighted tree-language $\mathcal{M}_i = (M_i, |.|_{a,i})$ where the $|.|_{a,i}$ are defined by induction on i:

$$\begin{aligned}
|.|_{a,0} &:= \emptyset, \\
|f\langle g_1, \dots, g_n\rangle|_{a,i+1} &:= |f| \circ_a \langle |g_1|_{a,i}, \dots, |g_n|_{a,i}\rangle, \\
|\varepsilon|_{a,i+1} &:= [a|1].
\end{aligned}$$

Again we show by induction that $|.|_{a,i} \subseteq |.|_{a,i+1}$. The case $|.|_{a,0} \subseteq |.|_{a,1}$ is trivial. Suppose that $|.|_{a,i-1} \subseteq |.|_{a,i}$. Let $g \in M_{i-1}$. If $g = \varepsilon$ then $|g|_{a,i} = |g|_{a,i-1} = [a|1]$. If $g = f\langle g_1, \dots, g_n\rangle$ for some $n \in \mathbb{N}$, $f \in L^{(n)}$, $g_1, \dots, g_n \in M_{i-1}$ then:

$$\begin{aligned}
|f\langle g_1, \dots, g_n\rangle|_{a,i+1} &= |f| \circ_a \langle |g_1|_{a,i}, \dots, |g_n|_{a,i}\rangle \\
&= |f| \circ_a \langle |g_1|_{a,i-1}, \dots, |g_n|_{a,i-1}\rangle \\
&= |f\langle g_1, \dots, g_n\rangle|_{a,i}.
\end{aligned}$$

From this follows at once that the inclusion function $\iota_i : M_i \rightarrowtail M_{i+1}$ is a homomorphism from $\mathcal{M}_i$ to $\mathcal{M}_{i+1}$ ($i \in \mathbb{N}$). Hence $(\mathcal{M}_i, \iota_i)_{i\in\mathbb{N}}$ is an injective ω-cochain.

Let us have a closer look onto the initial ω-cochain of the functor $\mathrm{S}_a(-, \mathcal{L})$. We will denote this chain by $(\mathcal{L}_i, \mu_i)_{i\in\mathbb{N}}$ where $\mathcal{L}_i$ and $\mu_i : \mathcal{L}_i \longrightarrow \mathcal{L}_{i+1}$ are given by:

$$\begin{aligned}
\mathcal{L}_0 &:= \emptyset, \\
\mathcal{L}_{i+1} &:= \mathrm{S}_a(\mathcal{L}_i, \mathcal{L}), \\
\mu_0 &:= \emptyset, \\
\mu_{i+1} &:= \mathrm{S}_a(\mu_i, \mathbf{1}_{\mathcal{L}}).
\end{aligned}$$

We will compare it with the injective cochain $(\mathcal{M}_i, \iota_i)_{i\in\mathbb{N}}$ which was defined previously.

Next we construct isomorphisms $\varphi_i : \mathcal{L}_i \longrightarrow \mathcal{M}_i$. The unique functions φ_0 and φ_1 are obviously homomorphisms since $|\varepsilon|_{a,1} = [a|1]$.

Suppose φ_n has already been constructed. Then

$$\begin{aligned}
\mathcal{L}_{n+1} = \mathcal{L} \cdot_a \mathcal{L}_n + \{[a|1]\} &\cong \left(\coprod_{f\in\mathcal{L}} |f| \cdot_a \mathcal{L}_i\right) + \{[a|1]\} \\
&\cong \left(\coprod_{k\in\mathbb{N}} \coprod_{f\in\mathcal{L}^{(k)}} |f| \cdot_a \mathcal{L}_n\right) + \{[a|1]\}
\end{aligned}$$

Recall also that $|f| \cdot_a \mathcal{L}_n = (L_n^k, |.|_f)$ where $|(t_1, \dots, t_k)|_f = |f| \circ_a \langle |t_1|, \dots, |t_k|\rangle$.

On the other hand

$$M_{n+1} = \bigcup_{k\in\mathbb{N}} \bigcup_{f\in\mathcal{L}^{(k)}} \{f\langle g_1, \dots, g_k\rangle \mid g_1, \dots, g_k \in M_n\} \cup \{\varepsilon\}.$$

Then we define $\varphi_{n+1} : \mathcal{L}_{n+1} \longrightarrow \mathcal{M}_{n+1}$. We start by defining on each of the direct summands (indexed by $f \in \mathcal{L}^{(k)}, k \in \mathbb{N}$) a function $\varphi_{n+1,f} : |f| \cdot_a \mathcal{L}_n \longrightarrow \mathcal{M}_{n+1}$

$$\varphi_{n+1,f} : (t_1, \dots, t_k) \mapsto f\langle \varphi_n(t_1), \dots, \varphi_n(t_k)\rangle.$$

Finally let $\kappa_{n+1} : \{[a|1]\} \longrightarrow \mathcal{M}_{n+1} : [a|1] \mapsto \varepsilon$. Now we define φ_{n+1} as the cotupling of all $\varphi_{n+1,f}$ $(f \in \mathcal{L})$ and of κ.

Since $|(t_1, \dots, t_k)|_f = |f| \circ_a \langle |t_1|, \dots, |t_k|\rangle$ and

$$\begin{aligned} |f\langle \varphi_n(t_1), \dots, \varphi_n(t_k)\rangle| &= |f| \circ_a \langle |\varphi_n(t_1)|, \dots, |\varphi_n(t_k)|\rangle \\ &= |f| \circ_a \langle |t_1|, \dots, |t_k|\rangle, \end{aligned}$$

we conclude that φ_{n+1} is indeed a homomorphism.

The bijectivity of the φ_i $(i \in \mathbb{N})$ is proved inductively. φ_0 is trivially bijective. Suppose φ_n is bijective. Let $t \in M_{n+1}$ then either $t = \varepsilon$ and $t = \kappa_{n+1}([a|1])$ or $t = f\langle g_1, \dots, g_k\rangle = f\langle \varphi_n(t_1), \dots, \varphi_n(t_k)\rangle = \varphi_{n+1,f}(t_1, \dots, t_n)$. Since φ_n is bijective, the choice for $t_1, \dots, t_n$ is unique. This proves bijectivity of φ_{n+1}.

Both cochains are injective—the first one by construction and the second one because $\mathrm{S}_a(-, \mathcal{L})$ preserves monos. Hence, by 2.21, we have that $(\mathcal{M}_i, \iota_i)_{i\in\mathbb{N}}$ and $(\mathcal{L}_i, \mu_i)_{i\in\mathbb{N}}$ are isomorphic. Therefore they have isomorphic colimits. $\square$

2.24 a-Annihilation. For $a \in \Sigma^{(0)}$ the functor $(- \cdot_a \emptyset) : \mathcal{L} \mapsto \mathcal{L}_{\neg a}$ is called *a-annihilation*. Since it is a special case of a-product, the a-annihilation preserves arbitrary colimits and monos.

2.25 Lemma. *Let $\mathcal{L} = (L, |.|) \in \mathsf{WTL}_\Sigma$. Define $(L_{\neg a}, |.|_{\neg a})$ where*

$$L_{\neg a} = \{s \in L \mid \mathrm{rk}_a(s) = 0\}$$

and $|.|_{\neg a}$ is the restriction of $|.|$ to $L_{\neg a}$. Then $(L_{\neg a}, |.|_{\neg a}) \cong \mathcal{L}_{\neg a}$.

Proof. Easy. $\square$

2.26 a-Recursion. Another, much more simple iteration operation may be obtained from the functor $\mathrm{R}_a : \mathsf{WTL}^2_\Sigma \longrightarrow \mathsf{WTL}_\Sigma : (X, L) \mapsto \mathcal{L} \cdot_a X$. The initial algebra carrier $\mu X.\mathrm{R}_a(X, -) \quad \mathcal{L} \mapsto \mathcal{L}^\mu_a$ is called the *a-recursion* of $\mathcal{L}$. The a-recursion is closely related to the a-iteration.

In order to reveal their connections we first introduce another functor

$$\mathrm{M}_a : \mathsf{WTL}^2_\Sigma \longrightarrow \mathsf{WTL}_\Sigma : (X, \mathcal{L}) \mapsto ((\mathcal{L} \cdot_a X) + \{[a|1]\})_{\neg a}.$$

2.27 Proposition. *For each $\mathcal{L} \in \mathsf{WTL}_\Sigma$ we have $\mathcal{L}^\mu_a \cong (\mathcal{L}^*_a)_{\neg a}$.*

Proof. We show by induction that the initial sequences of M_a and R_a are isomorphic. Let $(\mathcal{M}_i, \alpha_i)_{i\in\mathbb{N}}$, $(\mathcal{R}_i, \beta_i)_{i\in\mathbb{N}}$ be the respective initial sequences. Then $\mathcal{M}_0 = \mathcal{R}_0 = \emptyset$. Suppose we have already showed $\mathcal{M}_i \cong \mathcal{R}_i$. Then

$$\begin{aligned}
\mathcal{M}_{i+1} = M_a(\mathcal{M}_i) &= (\mathcal{L} \cdot_a \mathcal{M}_i + \{[a|1]\})_{\neg a} \\
&\cong (\mathcal{L} \cdot_a \mathcal{R}_i + \{[a|1]\})_{\neg a} \\
&\cong (\mathcal{L} \cdot_a \mathcal{R}_i)_{\neg a} + \{[a|1]\}_{\neg a} \\
&\cong (\mathcal{L} \cdot_a \mathcal{R}_i)_{\neg a} \\
&\cong \mathcal{L} \cdot_a \mathcal{R}_i \qquad\qquad \text{because of 2.16} \\
&= \mathcal{R}_{i+1}.
\end{aligned}$$

Since both sequences are obviously injective, by 2.21 they are isomorphic. Hence they have isomorphic colimits.

The rest follows from the fact that $[-]_{\neg a}$ preserves colimits. □

2.28 a-Semiiteration. For reasons of convenience, we introduce yet another iteration-operation:

$$\mathrm{P}_a : \mathsf{WTL}_\Sigma^2 \longrightarrow \mathsf{WTL}_\Sigma \qquad (X, \mathcal{L}) \mapsto \mathcal{L} \cdot_a (X + \{[a|1]\}).$$

The initial algebra-carrier $\mu X.\mathrm{P}_a(X, -)$ $\mathcal{L} \mapsto \mathcal{L}_A^+$ is called *a-semiiteration*. It is closely related to the a-iteration.

2.29 Lemma. *For $\mathcal{L} \in \mathit{WTL}_\Sigma$ we have $\mathcal{L}_a^* \cong \mathcal{L}_a^+ + \{[a|1]\}$.*

Proof. Let $(\mathcal{A}_i, \alpha_i)_{i\in\mathbb{N}}$ and $(\mathcal{B}_i, \beta_i)_{i\in\mathbb{N}}$ be the initial sequences of $\mathrm{P}_a(-, \mathcal{L})$ and of $\mathrm{S}_a(-, \mathcal{L})$, respectively. That is

$$\begin{aligned}
\mathcal{A}_0 &:= \emptyset & \mathcal{B}_0 &:= \emptyset \\
\mathcal{A}_{n+1} &:= \mathrm{P}_a(\mathcal{A}_n, \mathcal{L}) & \mathcal{B}_{n+1} &:= \mathrm{S}_a(\mathcal{B}_n, \mathcal{L}) \\
\alpha_0 &:= \emptyset & \beta_0 &:= \emptyset \\
\alpha_{n+1} &:= \mathrm{P}_a(\alpha_n, \mathbf{1}_\mathcal{L}) & \beta_{n+1} &:= \mathrm{S}_a(\beta_n, \mathbf{1}_\mathcal{L}).
\end{aligned}$$

We show that the sequences $(\mathcal{B}_{i+1}, \beta_{i+1})_{i\in\mathbb{N}}$ and $(\mathcal{A}_i + \{[a|1]\}, \alpha_i + \mathbf{1}_{\{[a|1]\}})_{i\in\mathbb{N}}$ are isomorphic. Obviously both sequences are injective.

For the induction-base we note that $\mathcal{B}_1 = \{[a|1]\}$ and $\mathcal{A}_0 + \{[a|1]\} = \emptyset + \{[a|1]\} = \{[a|1]\}$. We set $\varphi_0 := \mathbf{1}_{\{[a|1]\}}$.

Suppose now that the isomorphisms $\varphi_i : \mathcal{B}_{i+1} \longrightarrow \mathcal{A}_i + \{[a|1]\}$ exist for $i < n$. Then we define $\varphi_n : \mathcal{B}_{n+1} \longrightarrow \mathcal{A}_n + \{[a|1]\}$ as $\mathcal{L} \cdot_a \varphi_{n-1} + \mathbf{1}_{\{[a|1]\}} = \mathrm{S}_a(\varphi_{n-1}, \mathbf{1}_\mathcal{L})$. This is obviously an isomorphism. Hence, by 2.21, the sequences $(\mathcal{A}_i + \{[a|1]\}, \alpha_i + \mathbf{1}_{\{[a|1]\}})_{i\in\mathbb{N}}$ and $(\mathcal{B}_{i+1}, \beta_{i+1})_{i\in\mathbb{N}}$ are isomorphic. Consequently they have isomorphic colimits. Since the coproduct-functor preserves directed colimits, the first series has $\mathcal{L}_a^+ + \{[a|1]\}$ as colimit. The other series differs from the initial sequence of the functor $\mathrm{S}_a(-, \mathcal{L})$ only in the first element. Since this is equal to the empty language, it has no influence on the colimit. Hence the second series has $\mathcal{L}_a^*$ as colimit. □

2.30 Corollary. *For $\mathcal{L} \in \mathsf{WTL}_\Sigma$ we have $\mathcal{L}_a^\mu \cong (\mathcal{L}_a^+)_{\neg a}$.* □

2.31 Remark. Lemma 2.29 suggests another way to compute the a-semiiteration. Given $\mathcal{L} = (L, |.|) \in \mathsf{WTL}_\Sigma$. Let $(L_a^+, \mathrm{rk}, \circ) := (L, \mathrm{rk}_a)^+$. Define $|.|_a^+ : L_a^+ \longrightarrow \mathrm{WT}_\Sigma$ according to $|t|_a^+ := |t|$ for $t \in L$ and

$$|t\langle t_1, \ldots, t_{\mathrm{rk}_a(t)}\rangle|_a^+ := |t| \circ_a \langle |t_a|_a^+, \ldots, |t_n|_a^+\rangle.$$

Then $(L_a^+, |.|_a^+) \cong \mathcal{L}_a^+$.

2.32 Finitary and a-quasiregular languages. Let $\mathcal{L} = (L, |.|) \in \mathsf{WTL}_\Sigma$. $\mathcal{L}$ is called *finitary* if $\forall t \in \mathrm{T}_\Sigma$ the set $\{s \in L \mid \mathrm{ut}(|s|) = t\}$ is finite. Additionally we call $\mathcal{L}$ *a-quasiregular*[4] if it does not contain any element s with $\mathrm{ut}(|s|) = a$.

2.33 Proposition. *Let $\mathcal{L}_1, \ldots, \mathcal{L}_n$ be finitary weighted treelanguages. Let $c \in K$, $f \in \Sigma^{(n)}$, $a \in \Sigma^{(0)}$. Then the following weighted tree-languages are also finitary: $\mathcal{L}_1 + \mathcal{L}_2$, $c \cdot \mathcal{L}_1$, $[f|c]\langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle$, $\mathcal{L}_1 \cdot_a \mathcal{L}_2$ and $(\mathcal{L}_1)_{\neg a}$.* □

2.34 Proposition. *Let $\mathcal{L} \in \mathsf{WTL}_\Sigma$ be finitary. Then $\mathcal{L}_a^*$ is finitary if and only if $\mathcal{L}$ is a-quasiregular.*

Proof. Let $(L_a^*, |.|_a^*) \cong \mathcal{L}_a^*$ be given as in 2.23. Suppose there is an $s \in \mathcal{L}$ such that $\mathrm{ut}(|s|) = a$. Then $\mathrm{rk}_a(s) = 1$ and L_a^* contains s, $s\langle s\rangle$, $s\langle s\langle s\rangle\rangle, \ldots$ each of which has a as underlying tree. Hence $\mathcal{L}_a^*$ is not finitary.

Suppose $\mathcal{L}$ is finitary and a-quasiregular. Let $t \in \mathrm{T}_\Sigma$ with $\mathrm{size}_a(t) = k$. Since $\mathcal{L}$ is finitary, there are only finitely many $s \in \mathcal{L}$ such that $\mathrm{size}_a(|s|) \leq k$. Every element $r \in L_a^*$ with $\mathrm{ut}(|r|_a^*) = t$ has to be composed of these finitely many elements. Since $\mathcal{L}$ is a-quasiregular, by 1.13 there are just finitely many such compositions. Hence $\mathcal{L}_a^*$ contains only finitely many elements with underlying tree t. □

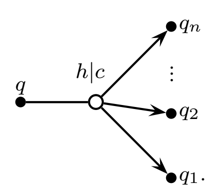
[4]The term "a-quasiregular" is inspired by the notion of quasiregularity on formal power-series. A formal power-series is called quasiregular, if the coefficient of the empty word is 0 (cf. [38]).

3 Weighted Tree-Automata

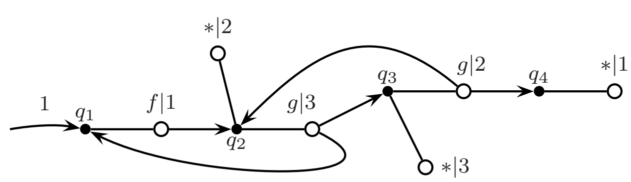
In this section we recapitulate the definition of weighted tree automata and show how they can be used to recognize weighted treelanguages. Moreover we shortly discuss the issue of top-down and bottom-up recognizability in our context. Note that our definition of weighted tree-automata slightly differs from the usual one because we allow a multiset of transitions instead of a set. This is done mainly for technical convenience. In particular it has no influence on the concept of recognizability.

3.1 Weighted tree-automata. Given a ranked alphabet Σ and a semiring K, a *finite weighted tree automaton* (WTA) is a quadruple $(Q, I, \iota, T, \lambda)$ such that Q is a finite set of states, $I \subseteq Q$ is a set of initial states, $\iota : I \longrightarrow K$ gives the initial weights, T is a finite ranked set of transition-symbols and λ is a function assigning to each transition-symbol $t \in T^{(n)}$ a transition $(q, f, q_1, \ldots, q_n, c)$ where $q, q_1, \ldots, q_n \in Q$, $c \in K$, $f \in \Sigma^{(n)}$ and $n \in \mathbb{N}$. For convenience we also define $\mathrm{lab}(t) := f$, $\mathrm{wt}(t) := c$, $\mathrm{dom}(t) := q$, $\mathrm{cod}_i(t) := q_i$ $(1 \leq i \leq n)$ and $\mathrm{cod}(t) := \{q_1, \ldots, q_n\}$.

3.2 Example. Let K be the semiring of natural numbers. $\Sigma^{(0)} = \{*\}$, $\Sigma^{(1)} = \{f\}$, $\Sigma^{(2)} = \{g\}$, $\mathcal{A} = (Q, I, \iota, T, \lambda)$ where $Q = \{q_1, \ldots, q_4\}$, $I = \{q_1\}$, $\iota(q_1) = 1$, $T = \{t_1, \ldots, t_6\}$, $\lambda(t_1) = (q_1, f, q_2, 1)$, $\lambda(t_2) = (q_2, *, 2)$, $\lambda(t_3) = (q_2, g, q_1, q_3, 3)$, $\lambda(t_4) = (q_3, *, 3)$, $\lambda(t_5) = (q_3, g, q_4, q_2, 2)$, $\lambda(t_6) = (q_4, *, 1)$. Since such a description is very tedious, we also give a pictorial representation of $\mathcal{A}$. Note that in this representation a transition-symbol t with $\lambda(t) = (q, h, q_1, \ldots, q_n, c)$ is depicted as



The output-edges are always ordered counterclockwise starting immediately left of the input-edge. The initial costs are depicted by arrows to the initial states carrying costs:

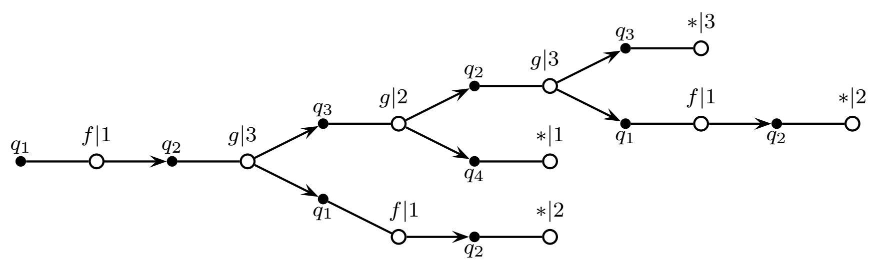


Our graphical representation of WTAs is similar to a graphical representation of bottom-up tree-automata by Petri-nets that was proposed by Reisig [45].

WTAs read trees and output weights. In order to give a strict definition of how a WTA processes a tree, we introduce the notion of a run along a tree through a WTA:

3.3 Runs. *Runs* through $\mathcal{A}$ along trees are defined inductively: If $a \in \Sigma^{(0)}$ and $\tau \in T$ such that $\lambda(\tau) = (q, a, c)$ for some $q \in Q$ and $c \in K$, then τ is a run along a with root q. If $f \in \Sigma^{(n)}$ and $t_1, \ldots, t_n \in \mathrm{T}_\Sigma$ with runs $p_1, \ldots, p_n$ rooting in states $q_1, \ldots, q_n$, respectively, and if $\tau \in T$ with label $\lambda(\tau) = (q, f, q_1, \ldots, q_n, c)$ then $\tau\langle p_1, \ldots, p_n\rangle$ is a run along $f\langle t_1, \ldots t_n\rangle$ with root q. A run is called *initial* if its root is an initial state. With $\mathrm{run}(t)$ we denote the set of all initial runs along t and with $\mathrm{run}(\mathcal{A})$ we denote the set of all initial runs through $\mathcal{A}$.

3.4 Example. Let $\mathcal{A}$ be the WTA from 3.2. Then $t_1\langle t_3\langle t_1\langle t_2\rangle, t_5\langle t_6, t_3\langle t_1\langle t_2\rangle, t_4\rangle\rangle\rangle\rangle$ is an initial run through $\mathcal{A}$ along the tree $f\langle g\langle f\langle *\rangle, g\langle *, g\langle f\langle *\rangle, *\rangle\rangle\rangle\rangle$. In a more convenient pictorial way this is:



3.5 Address-based description of runs. The address-based description of trees and weighted trees (cf. 1.6) suggests also to give an address-based description of runs. Let $t \in \mathrm{T}_\Sigma$ such that $t = (\mathrm{adr}(t), \mathrm{lab}_t)$ and let p be a run of $\mathcal{A}$ along t. Then p may be described by the pair $(\mathrm{adr}(t), \mathrm{trans}_p)$ where $\mathrm{trans}_p : \mathrm{adr}(t) \longrightarrow T$. The precise definition of trans_p is done by induction on the structure of t. If $t = a$ and $p = \tau$ then $\mathrm{adr}(t) = \{\varepsilon\}$ and we define $\mathrm{trans}_p(\varepsilon) := \tau$. If $t = f\langle t_1, \ldots, t_n\rangle$ and hence $p = \tau\langle p_1, \ldots, p_n\rangle$, then $\mathrm{adr}(t) = \{\varepsilon\} \cup \bigcup_{i=1}^n i \cdot \mathrm{adr}(t_i)$ and we define $\mathrm{trans}_p(\varepsilon) := \tau$ and $\mathrm{trans}_p(i \cdot w) := \mathrm{trans}_{p_i}(w)$. It is fairly easy to see that the pair $(\mathrm{adr}(t), \mathrm{trans}_p)$ characterizes p completely.

3.6 Lemma. *Given a tree* $t \in \mathrm{T}_\Sigma$ *with address-based description* $(\mathrm{adr}(t), \mathrm{lab}_t)$. *Let* $\mathrm{trans} : \mathrm{adr}(t) \longrightarrow T$. *Then* $(\mathrm{adr}(t), \mathrm{trans})$ *describes a run if and only if*

1. $\forall w \in \mathrm{adr}(t) : \mathrm{rk}(\mathrm{lab}_t(w)) = \mathrm{rk}(\mathrm{trans}(w))$,

2. $\forall w \cdot i \in \mathrm{adr}(t) : \mathrm{cod}_i(\mathrm{trans}(w)) = \mathrm{dom}(\mathrm{trans}(w \cdot i))$.

$\square$

3.7 WTL-semantics of WTAs, recognizability. To each run p we may associate a weighted tree $|p|$: If p consists just of one transition $\tau \in T$ and $\lambda(\tau) = (q, a, c)$, then we define $|p| := [a|c]$. In the composite case where $p = \tau\langle p_1, \dots , p_n\rangle$ where $\lambda(\tau) = (q, f, q_1, \dots , q_n, c)$ we define $|p| := [f|c]\langle |p_1|, \dots , |p_n|\rangle$.

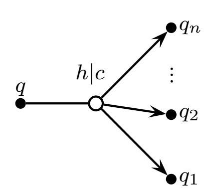
This definition allows us to associate with each WTA $\mathcal{A}$ a weighted tree-language $\mathcal{L}_{\mathcal{A}} : \mathrm{run}(\mathcal{A}) \longrightarrow \mathrm{WT}_\Sigma$ where an initial run p with root q is mapped to $\iota(q) \cdot |p|$. Two WTAs $\mathcal{A}_1$ and $\mathcal{A}_2$ are called *equivalent* (denoted by $\mathcal{A}_1 \equiv \mathcal{A}_2$) if their weighted tree-languages are isomorphic. A weighted tree-language $\mathcal{L}$ is called *recognizable* if there is a finite WTA $\mathcal{A}$ with $\mathcal{L}_{\mathcal{A}} \cong \mathcal{L}$.

3.8 Example. The weighted tree associated with the run from Example 3.4 is the one from Example 1.5

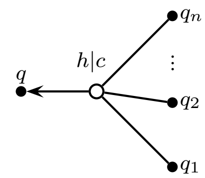
3.9 Remark. Our definition of WTAs is a bit unusual as we allow a finite multiset of transitions. This is essential in the realm of weighted tree-languages as without it some of our constructions on WTAs will not work or will not yield the desired results on the level of weighted tree-languages. However, when we use WTAs to recognize formal tree-series, then the existence or non-existence of multiple transitions will be of no importance anymore (multiple transitions can be combined to one by adding their weights). In particular our WTAs will recognize precisely the recognizable formal tree-series in the classical sense.

3.10 Remark. Recognizable weighted tree-languages have the special property to be finitary. Therefore it is easy to see, that the class of recognizable weighted tree-languages is not closed with respect to a-iteration (e.g $\{[a|c]\}_a^*$ is not recognizable, cf. also 2.34). We will see later on that this problem may be solved by allowing the application of a-iteration only to a-quasiregular weighted tree-languages.

3.11 Top-down/bottom-up issue. In contrast to other authors (cf. [8]) we do not distinguish between bottom-up and top-down weighted tree-automata. The reason is that in our setting the difference between the two principles is purely pictorial. In Example 3.2 we chose to draw transition symbols with one input and n outputs



and our WTAs have initial states and initial costs. Thus we created the intuition that the automata read trees top-down. Had we reversed the direction of the edges like



and had we called I "set of final states" and their weights "final weights", then the automaton would seem to read the trees bottom-up. However, this has neither an influence on the formal representation of WTAs nor on the definition of runs. Consequently it also has no influence on the recognized weighted tree-languages.

Note that a difference arises if we are dealing with questions of determinism. As usual bottom-up- and top-down-determinizability are not equivalent. But such problems are of no interest in this thesis.

4 Weak Weighted Tree-Automata

WTAs are a rather straight forward generalization of weighted automata on words. We would like to study the connections between rational expressions (to be defined later in our context) and recognizable weighted tree-languages. To this end we need to extend the notion of WTAs slightly by introducing silent transitions. We call the new type of automata *weak weighted tree automata* (wWTA). Weighted tree languages recognized by wWTAs we call weakly recognizable. Indeed it turns out that the class of weakly recognizable weighted tree languages is strictly larger than the class of recognizable weighted tree languages because the first class contains infinitary languages. We characterize those wWTAs that recognize finitary languages and go on with a necessary and sufficient criterion when the language recognized by a wWTA is recognizable. This characterization corresponds to the problem of ε-removal in classical automata theory. An immediate consequence of these results is that in fact a weakly recognizable weighted tree-language is recognizable if and only if it is finitary.

A big part of this section is spent for the introduction of several operations on wWTAs such as topcatenation, a-product, a-recursion etc. For each such operation a close relation to the corresponding counterpart on weighted tree languages is demonstrated. As immediate consequence we get that the weakly recognizable weighted tree-languages are preserved under all our operations.

At the very end we show that the a-iteration of a recognizable weighted tree language is recognizable if and only if the the original weighted tree-language is a-quasiregular.

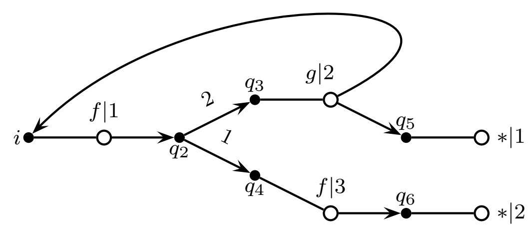
4.1 Weak weighted tree-automata. Let Σ be a rank-alphabet and K be a semiring. A weak *weighted tree-automaton* (wWTA) is a tuple $(Q, i, T, \lambda, S, \sigma)$ where

1. Q is a finite state-set,
2. $i \in Q$ is an initial state,
3. T is a finite set of transition-symbols,
4. λ is a function that assigns to each transition-symbol a transition $(q, f, q_1, \dots, q_n, c)$ where $f \in \Sigma^{(n)}$, $q, q_1, \dots, q_n$ are states, $c \in K$ and $n \in \mathbb{N}$,
5. S is a finite set of silent transition-symbols,
6. σ assigns to each silent transition-symbol a silent transition (q_1, q_2, c) where $q_1, q_2 \in Q$, $c \in K$

such that $Q \cap T = Q \cap S = T \cap S = \emptyset$.

The functions lab, wt, dom, cod_i, cod for transition-symbols from T are defined like in 3.1. If $s \in S$ with $\sigma(s) = (q_1, q_2, c)$ we define additionally $\mathrm{dom}(s) := q_1$, $\mathrm{cod}(s) := q_2$ and $\mathrm{wt}(s) := c$.

4.2 Example. Next we give an example of a wWTA. As usual K is assumed to be the semiring of natural numbers. $\Sigma = \{f, g, *\}$, $\mathrm{rk}(f) = 1$, $\mathrm{rk}(g) = 2$, $\mathrm{rk}(*) = 0$, $\mathcal{A} = (Q, i, T, \lambda, S, \sigma)$, $Q = \{i, q_2, \ldots, q_6\}$, $T = \{t_1, \ldots, t_5\}$, $\lambda(t_1) = (i, f, q_2, 1)$, $\lambda(t_2) = (q_3, g, q_5, i, 2)$, $\lambda(t_3) = (q_4, f, q_6, 3)$, $\lambda(t_4) = (q_5, *, 1)$, $\lambda(t_5) = (q_6, *, 2)$, $S = \{s_1, s_2\}$, $\sigma(s_1) = (q_2, 2, q_3)$, $\sigma(s_2) = (q_2, 1, q_4)$. Like in Example 3.2 we prefer the pictorial presentation of wWTAs. Transition-symbols are depicted as usual. The silent transition-symbols are represented by arrows that are equipped with a weight:



4.3 WTL-semantics of wWTAs, weak recognizability. Just like WTAs, weak WTAs are meant to define (or recognize) weighted tree-languages. Indeed, the dynamics of weak WTAs is almost the same the the one of WTAs. First we define runs through wWTAs and then we associate with each run a weighted tree. In the following, if not said otherwise, let $\mathcal{A} = (Q, i, T, \lambda, S, \sigma)$ be a wWTA.

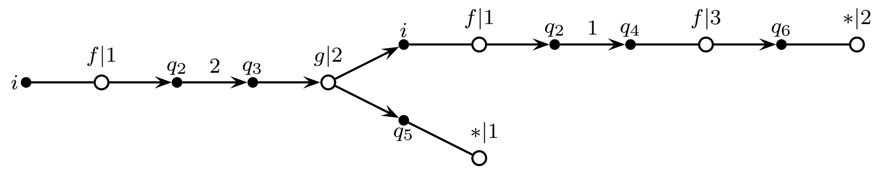
Runs through $\mathcal{A}$ along trees are defined inductively: If $a \in \Sigma_0$ and $\tau \in T$ with $\lambda(\tau) = (q, a, c)$, then τ is a run of $\mathcal{A}$ along the tree a with root q. If p is a run of $\mathcal{A}$ with root q along some tree and if $s \in S$ with $\sigma(s) = (q', q, c)$, then $s \cdot p$ is a run of $\mathcal{A}$ with root q' along the same tree. If $f \in \Sigma^{(n)}$, for $1 \le i \le n$, $t_i \in \mathrm{T}_\Sigma$, p_i is a run of $\mathcal{A}$ along t_i rooting in q_i and $\tau \in T$ with $\lambda(\tau) = (q, f, q_1, \ldots, q_n)$, then $\tau\langle p_1, \ldots, p_n\rangle$ is a run of $\mathcal{A}$ along $f\langle t_1, \ldots, t_n\rangle$ with root q.

A run of $\mathcal{A}$ is called *initial run* if its root is i. For $t \in \mathrm{T}_\Sigma$ we denote by $\mathrm{run}(t)$ the set of all initial runs of $\mathcal{A}$ along t. Finally we define $\mathrm{run}(\mathcal{A})$ as the set of all initial runs in $\mathcal{A}$.

Next we associate to each run p of $\mathcal{A}$ a weighted tree $|p|$: If $p = \tau$ for some $\tau \in T$ and if $\lambda(\tau) = (q, a, c)$, then $|p| := [a|c]$. If $p = s \cdot p'$ and $\sigma(s) = (q', q, c)$, then $|p| := c \cdot |p'|$. If $p = \tau\langle p_1, \ldots, p_n\rangle$ and $\lambda(\tau) = (q, f, q_1, \ldots, q_n, c)$ then $|p| := [f|c]\langle |p_1|, \ldots, |p_n|\rangle$.

As before we define the weighted tree-language $\mathcal{L}_\mathcal{A}$ associated to $\mathcal{A}$ according to $\mathcal{L}_\mathcal{A} : \mathrm{run}(\mathcal{A}) \longrightarrow \mathrm{WT}_\Sigma$, $p \mapsto |p|$. A weighted tree language $\mathcal{L}$ that is isomorphic to $\mathcal{L}_\mathcal{A}$ for some finite wWTA $\mathcal{A}$, is called *weakly recognizable*. Again two wWTAs $\mathcal{A}_1$ and $\mathcal{A}_2$ will be called equivalent if they recognize isomorphic weighted tree-languages. In this case we will write $\mathcal{A}_1 \equiv \mathcal{A}_2$.

4.4 Example. Let $\mathcal{A}$ be the wWTA from 4.2. Then $t_1\langle s_1 \cdot t_2\langle t_4, t_1\langle s_2 \cdot t_3\langle t_5\rangle\rangle\rangle\rangle$ is a run through $\mathcal{A}$ along $f\langle g\langle *, f\langle f\langle *\rangle\rangle\rangle\rangle$. Pictorially this run looks as follows:



4.5 Proposition. *Every recognizable weighted tree-language is weakly recognizable.*

Proof. Let $\mathcal{L}$ be the weighted tree-language recognized by the weighted tree-automaton $\mathcal{A} = (Q, I, \iota, T\,\lambda)$. Let $i \notin Q$ and $S = (s_q)_{q\in I}$ be a family of silent transition-symbols. Define σ by

$$\sigma\,:\,s_q \mapsto (i, q, \iota(q)).$$

Then the weighted tree-language that is recognized by the weak weighted tree-automaton $\mathcal{A}' = (Q \cup \{i\}, i, T, \lambda, S, \sigma)$ obviously is isomorphic to $\mathcal{L}$. $\square$

4.6 Silent paths, silent cycles. A word $\mathbf{s} = s_1 \cdots s_k \in S^*$ of silent transitions of $\mathcal{A}$ is called *silent path* if $\operatorname{cod}(s_i) = \operatorname{dom}(s_{i+1})$ $(1 \le i < k)$. By convention, the empty word ε counts also as a silent path. We may extend dom and cod to non-empty silent paths according to $\operatorname{dom}(\mathbf{s}) := \operatorname{dom}(s_1)$, $\operatorname{cod}(\mathbf{s}) := \operatorname{cod}(s_k)$. A silent path $\mathbf{s}$ with $\operatorname{dom}(\mathbf{s}) = \operatorname{cod}(\mathbf{s})$ is called *silent cycle*. The set of all silent paths of $\mathcal{A}$ is denoted by $\mathrm{sP}_{\mathcal{A}}$.

To each silent path $\mathbf{s} \in \mathrm{sP}_{\mathcal{A}}$ we assign a weight $\mathrm{wt}(\mathbf{s}) \in K$ according to $\mathrm{wt}(\varepsilon) := 1$, $\mathrm{wt}(s \cdot \mathbf{s}) := \mathrm{wt}(s) \odot \mathrm{wt}(\mathbf{s})$.

Silent cycles will play a crucial role in the characterization of the finitary weakly recognizable weighted tree-languages.

4.7 Address-based description of runs. The address-based description of trees and weighted trees from 1.6 allows us to give a non-inductive description of runs of $\mathcal{A}$: Given a run p of $\mathcal{A}$ we define $\mathrm{trans}_p : \mathrm{adr}(|p|) \longrightarrow T$ and $\mathrm{spath}_p : \mathrm{adr}(|p|) \longrightarrow \mathrm{sP}_{\mathcal{A}}$. As usually the definition of these function is done by induction on the structure of p.

If $p = \tau \in T$ with $\lambda(\tau) = (q, a, c)$, then $\mathrm{adr}(|p|) = \{\varepsilon\}$ and we define $\mathrm{trans}_p(\varepsilon) := \tau$ and $\mathrm{spath}_p(\varepsilon) := \varepsilon$.

If $p = s \cdot p'$ where $s \in S$ then $\mathrm{adr}(p) = \mathrm{adr}(p')$ and we define $\mathrm{trans}_p(\varepsilon) := \mathrm{trans}_{p'}(\varepsilon)$, $\mathrm{spath}_p(\varepsilon) := s \cdot \mathrm{spath}_{p'}(\varepsilon)$ and for $w \neq \varepsilon$: $\mathrm{trans}_p(w) := \mathrm{trans}_{p'}(w)$, $\mathrm{spath}_p(w) := \mathrm{spath}_{p'}(w)$.

If $p = \tau\langle p_1, \ldots, p_n\rangle$ where $\tau \in T$ and $\lambda(\tau) = (q, f, q_1, \ldots, q_n, c)$, then $\mathrm{adr}(|p|) = \{\varepsilon\} \cup \bigcup_{i=1}^{n}\{i \cdot w \mid w \in \mathrm{adr}(|p_i|)\}$ and we define $\mathrm{trans}_p(\varepsilon) := \tau$, $\mathrm{spath}_p(\varepsilon) := \varepsilon$ and $\mathrm{trans}_p(i \cdot w) := \mathrm{trans}_{p_i}(w)$, $\mathrm{spath}_p(i \cdot w) := \mathrm{spath}_{p_i}(w)$.

Clearly, every run p of $\mathcal{A}$ is determined by the triple $(\mathrm{adr}(|p|), \mathrm{trans}_p, \mathrm{spath}_p)$.

4.8 Reduced wWTAs, reachability. Let $\mathcal{A} = (Q, i, T, \lambda, S, \sigma)$ be a wWTA and let $q \in Q$. A transition-symbol $\tau \in T$ is called *reachable from* q if there is a run p of $\mathcal{A}$ with root q and with address-based description $(\mathrm{adr}(|p|), \mathrm{trans}_p, \mathrm{spath}_p)$ such that for some $w \in \mathrm{adr}(|p|)$ we have $\mathrm{trans}_p(w) = \tau$. A silent transition-symbol $s \in S$ is called *reachable in* q if there is a run p of $\mathcal{A}$ rooting in q with address-based description $(\mathrm{adr}(|p|), \mathrm{trans}_p, \mathrm{spath}_p)$ such that for some $w \in \mathrm{adr}(|p|)$ there are silent paths $\mathbf{s}_1$ and $\mathbf{s}_2$ with $\mathrm{spath}_p(w) = \mathbf{s}_1 s \mathbf{s}_2$. If any silent transition of a silent cycle is *reachable* then the cycle is called reachable. A state $q' \in Q$ is called *reachable from* q if there is a transition-symbol $x \in T \cup S$ that is reachable from q such that $\mathrm{dom}(x) = q'$. If q is the initial state of $\mathcal{A}$ then instead of “reachable from q” we usually say just “reachable”. The wWTA $\mathcal{A}$ is called *reduced* if all states of $\mathcal{A}$ are reachable. Note that this implies that also all (silent) transition-symbols of $\mathcal{A}$ are reachable.

4.9 Lemma. *Let p be a run of $\mathcal{A}$ with description* $(\mathrm{adr}(|p|), \mathrm{trans}_p, \mathrm{spath}_p)$*. Suppose that the weighted tree $|p|$ is described by* $(\mathrm{adr}(|p|), \mathrm{lab}_{|p|}, \mathrm{wt}_{|p|})$*. Then*

1. $\mathrm{lab}_{|p|}(w) = \mathrm{lab}(\mathrm{trans}_p(w))$,
2. $\mathrm{wt}_{|p|}(w) = \mathrm{wt}(\mathrm{spath}_p(w)) \odot \mathrm{wt}(\mathrm{trans}_p(w))$.

Proof. We proceed by induction on the structure of p:

If $p = \tau \in T$, then

$$\mathrm{lab}_{|p|}(\varepsilon) = \mathrm{lab}(\tau) = \mathrm{lab}(\mathrm{trans}_p(\varepsilon))$$

and

$$\begin{aligned}\mathrm{wt}_{|p|}(\varepsilon) &= 1 \odot \mathrm{wt}(\tau)\\ &= \mathrm{wt}(\varepsilon) \odot \mathrm{wt}(\mathrm{trans}_p(\varepsilon))\\ &= \mathrm{wt}(\mathrm{spath}_p(\varepsilon)) \odot \mathrm{wt}(\mathrm{trans}_p(\varepsilon)).\end{aligned}$$

If $p = s \cdot p'$, $s \in S$ then since $|p| = \mathrm{wt}(s) \odot |p'|$:

$$\mathrm{lab}_{|p|}(w) = \mathrm{lab}_{|p'|}(w) \qquad \text{and}$$
$$\mathrm{wt}_{|p|}(w) = \begin{cases} \mathrm{wt}_{|p'|}(w) & w \neq \varepsilon \\ \mathrm{wt}(s) \odot \mathrm{wt}_{|p'|}(w) & \text{else.} \end{cases}$$

Because of this and because $\mathrm{trans}_p = \mathrm{trans}_{p'}$ and spath_p, $\mathrm{spath}_{p'}$ only differ at ε, the claims hold for all $w \neq \varepsilon$. For the remaining case $w = \varepsilon$ we argue

$$\mathrm{lab}_{|p|}(\varepsilon) = \mathrm{lab}_{|p'|}(\varepsilon) = \mathrm{lab}(\mathrm{trans}_{p'}(\varepsilon)) = \mathrm{lab}(\mathrm{trans}_p(\varepsilon))$$

and

$$\begin{aligned}\mathrm{wt}_{|p|}(\varepsilon) &= \mathrm{wt}(s) \odot \mathrm{wt}_{|p'|}(\varepsilon)\\ &= \mathrm{wt}(s) \odot \mathrm{wt}(\mathrm{spath}_{p'}(\varepsilon)) \odot \mathrm{wt}(\mathrm{trans}_{p'}(\varepsilon))\\ &= \mathrm{wt}(\mathrm{spath}_p(\varepsilon)) \odot \mathrm{wt}(\mathrm{trans}_p(\varepsilon)).\end{aligned}$$

If $p = \tau\langle p_1, \ldots, p_n\rangle$, $\tau \in T$, then $|p| = [\mathrm{lab}(\tau)|\,\mathrm{wt}(\tau)]\langle |p_1|, \ldots, |p_n|\rangle$ and hence

$$\begin{aligned}\mathrm{lab}_{|p|}(\varepsilon) &= \tau = \mathrm{lab}(\mathrm{trans}_p(\varepsilon))\\ \mathrm{wt}_{|p|}(\varepsilon) &= \mathrm{wt}(\tau) = 1 \odot \mathrm{wt}(\mathrm{trans}_p(\varepsilon)) = \mathrm{wt}(\mathrm{spath}_p(\varepsilon)) \odot \mathrm{wt}(\mathrm{trans}_p(\varepsilon)))\end{aligned}$$

and

$$\begin{aligned}\mathrm{lab}_{|p|}(i \cdot w) &= \mathrm{lab}_{|p_i|}(w) = \mathrm{lab}(\mathrm{trans}_{p_i}(w))\\ &= \mathrm{lab}(\mathrm{trans}_p(i \cdot w)),\end{aligned}$$

$$\begin{aligned}\mathrm{wt}_{|p|}(i \cdot w) &= \mathrm{wt}_{|p_i|}(w)\\ &= \mathrm{wt}(\mathrm{spath}_{p_i}(w)) \odot \mathrm{wt}(\mathrm{trans}_{p_i}(w))\\ &= \mathrm{wt}(\mathrm{spath}_p(w)) \odot \mathrm{wt}(\mathrm{trans}_p(i \cdot w)).\end{aligned}$$

□

4.10 Lemma. *Let* $t \in \mathrm{T}_\Sigma$ *with address-based description* $(\mathrm{adr}(t), \mathrm{lab}_t)$. *Further on let* $\mathrm{trans} : \mathrm{adr}(t) \longrightarrow T$ *and* $\mathrm{spath} : \mathrm{adr}(t) \longrightarrow \mathrm{sP}_{\mathcal{A}}$. *Then* $(\mathrm{adr}(t), \mathrm{trans}, \mathrm{spath})$ *is a run of* $\mathcal{A}$ *if and only if*

1. *for all* $w \in \mathrm{adr}(t) : \mathrm{rk}(\mathrm{lab}_t(w)) = \mathrm{rk}(\mathrm{trans}(w))$,
2. *for all* $w \cdot i \in \mathrm{adr}(t) : \mathrm{cod}_i(\mathrm{trans}(w)) = \begin{cases}\mathrm{dom}(\mathrm{spath}(w \cdot i)) & \mathrm{spath}(w \cdot i) \neq \varepsilon\\ \mathrm{dom}(\mathrm{trans}(w \cdot i)) & \textit{else.}\end{cases}$

□

4.11 Lemma. *Let* $\mathcal{A}$ *be a* wWTA. *Then* $\mathcal{L}_{\mathcal{A}}$ *is finitary if and only if* $\mathcal{A}$ *does not contain a reachable silent cycle.*

Proof. Suppose $\mathcal{A}$ contains a reachable silent cycle $s_1 \cdots s_k$. Let p be an initial run of $\mathcal{A}$ that contains s_1 (such a run exists because the cycle is reachable). By 4.7 p may be described by the triple $(\mathrm{adr}(|p|), \mathrm{trans}_p, \mathrm{spath}_p)$. Since s_1 is on p, there exists an address $w \in \mathrm{adr}(|p|)$ such that $\mathrm{spath}_p(w) = \mathbf{s} \cdot s_1 \cdot \mathbf{s}'$. We define paths p_i ($i \in \mathbb{N}$) by giving their descriptions $(\mathrm{adr}(|p_i|), \mathrm{trans}_{p_i}, \mathrm{spath}_{p_i})$. In particular we set $\mathrm{adr}(|p_i|) := \mathrm{adr}(|p|)$, $\mathrm{trans}_{p_i} := \mathrm{trans}_p$ and

$$\mathrm{spath}_{p_i}(v) := \begin{cases}\mathrm{spath}_p(v) & v \neq w,\\ \mathbf{s} \cdot (s_1 \cdots s_k)^i \cdot \mathbf{s}' & \text{else.}\end{cases}$$

Using 4.10 we can see that indeed each such description corresponds to a run of $\mathcal{A}$. Moreover, all p_i have the same underlying tree as p but each run contains a different amount of silent transitions. Hence $\mathcal{L}_{\mathcal{A}}$ is not finitary.

Assume now, $\mathcal{L}_{\mathcal{A}}$ is infinitary. Let $t \in \mathrm{T}_\Sigma$ such that

$$\mathcal{L}_t = \{p \in \mathcal{L}_{\mathcal{A}} \mid \mathrm{ut}(|p|) = t\}$$

is infinite. All the elements of $\mathcal{L}_t$ share the same set of addresses – namely $\operatorname{adr}(t)$. Since T is finite, there are only finitely many mappings from $\operatorname{adr}(t)$ to T. Let $S_t := \{\mathbf{s} \in \mathrm{sP}_{\mathcal{A}} \mid \exists p \in \mathcal{L}_t, \exists v \in \operatorname{adr}(t) : \operatorname{spath}_p(v) = \mathbf{s}\}$. Then S_t is infinite. On the other hand there are only finitely many silent transitions. Hence S_t contains arbitrarily long words. If the length of a silent path exceeds the number of states of $\mathcal{A}$, then it must contain a cyclic subpath (which is reachable by construction). Hence $\mathcal{A}$ contains a reachable silent cycle. □

4.12 Proposition. *Let $\mathcal{A}$ be a* wWTA *without reachable silent cycles. Then there is a* WTA *$\mathcal{A}'$ such that $\mathcal{L}_{\mathcal{A}} \cong \mathcal{L}_{\mathcal{A}'}$.*

Proof. Let $\mathcal{A} = (Q, i, T, \lambda, S, \sigma)$ be a wWTA. Without loss of generality we may assume that all its silent transitions are reachable because otherwise the non-reachable ones may be removed without altering the weighted tree-language recognized by $\mathcal{A}$. Assume further on that $\mathcal{A}$ does not contain silent cycles. Then we claim that $\mathrm{sP}_{\mathcal{A}}$ is finite, for assume it is not, then it contains words of arbitrary length (because S is finite). Hence it would also contain a word of length $> |Q|$ but such a word contains necessarily a cycle – contradiction.

Let us construct the automaton $\mathcal{A}'$ now. Its state set be Q and we define $I = \{i\}$, $\iota(i) := 1$. The set T' of transition-symbols of $\mathcal{A}'$ is defined as follows:

$$T' := \{(\mathbf{s}, t) \mid \mathbf{s} \in \mathrm{sP}_{\mathcal{A}},\, t \in T,\, \mathbf{s} = \varepsilon \text{ or } \operatorname{cod}(\mathbf{s}) = \operatorname{dom}(t)\}$$

and $\lambda'(\mathbf{s}, t) := (q', f, q_1, \ldots, q_n, c')$ where $\lambda(t) = (q, f, q_1, \ldots, q_n, c)$ and where $c' := \operatorname{wt}(\mathbf{s}) \odot c$ and

$$q' = \begin{cases} q & \text{if } \mathbf{s} = \varepsilon \\ \operatorname{dom}(\mathbf{s}) & \text{else.} \end{cases}$$

Altogether $\mathcal{A}' = (Q, I, \iota, T', \lambda')$. It remains to show that $\mathcal{A} \equiv \mathcal{A}'$. For this we will use the address-based characterization of runs from 3.5 and 4.7. Next we define $\varphi : \operatorname{run}(\mathcal{A}) \longrightarrow \operatorname{run}(\mathcal{A}')$: Let $p = (\operatorname{adr}(|p|), \operatorname{trans}_p, \operatorname{spath}_p)$ be the description of a run of $\mathcal{A}$. We define $\varphi(p) := p' := (\operatorname{adr}(|p|), \operatorname{trans}_{p'})$ according to

$$\operatorname{trans}_{p'}(w) := (\operatorname{spath}_p(w), \operatorname{trans}_p(w)).$$

Let us see whether φ is well-defined: $(\operatorname{spath}_p(w), \operatorname{trans}_p(w))$ is obviously an element of T' by construction of spath_p and trans_p. Let $w \cdot i \in \operatorname{adr}(|p|)$. Then

$$\begin{aligned} \operatorname{cod}_i(\operatorname{trans}_{p'}(w)) &= \operatorname{cod}_i(\operatorname{trans}_p(w)) \\ &= \begin{cases} \operatorname{dom}(\operatorname{spath}_p(w \cdot i)) & \text{if } \operatorname{spath}_p(w \cdot i) \neq \varepsilon \\ \operatorname{dom}(\operatorname{trans}_p(w \cdot i)) & \text{else} \end{cases} \\ &= \operatorname{dom}(\operatorname{trans}_{p'}(w \cdot i)) \end{aligned}$$

Hence, by 3.6, p' is indeed a run of $\mathcal{A}'$. It remains to show that $|p| = |p'|$. That is, with $|p| = (\mathrm{adr}(|p|), \mathrm{lab}_{|p|}, \mathrm{wt}_{|p|})$ and $|p'| = (\mathrm{adr}(|p|), \mathrm{lab}_{|p'|}, \mathrm{wt}_{|p'|})$ we have to show that $\mathrm{lab}_{|p|} = \mathrm{lab}_{|p'|}$ and $\mathrm{wt}_{|p|} = \mathrm{wt}_{|p'|}$: However, by 4.9 we have

$$\mathrm{lab}_{|p|}(w) = \mathrm{lab}(\mathrm{trans}_p(w)) = \mathrm{lab}(\mathrm{trans}_{p'}(w)) = \mathrm{lab}_{|p'|}(w)$$

and

$$\mathrm{wt}_{|p|}(w) = \mathrm{wt}(\mathrm{spath}_p(w)) \odot \mathrm{wt}(\mathrm{trans}_p(w)) = \mathrm{wt}(\mathrm{trans}_{p'}(w)) = \mathrm{wt}_{|p'|}(w).$$

Hence φ is a homomorphism from $\mathcal{L}_{\mathcal{A}}$ to $\mathcal{L}_{\mathcal{A}'}$. Obviously, the construction of p' out of p may be reversed by setting $\mathrm{spath}_p := e_1^2 \circ \mathrm{trans}_{p'}$ and $\mathrm{trans}_p := e_2^2 \circ \mathrm{trans}_{p'}$ where e_1^2 and e_2^2 are the binary projections. Hence φ is bijective and thus it is an isomorphism. □

4.13 Remark. Our silent transition-symbols are inspired by the so called ε-transitions from classical automata-theory. These are transitions that do not read anything from the input word (or more precisely, they read the empty word ε). Introducing and eliminating such transitions is a technique that is widely used in automata-theory.

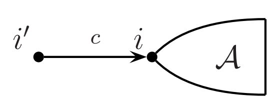
4.14 Corollary. *A weighted tree-language is recognizable if and only if it is finitary and weakly recognizable.*

Proof. Let $\mathcal{L}_{\mathcal{A}}$ be a weakly recognizable weighted tree-language that is recognized by the wWTA $\mathcal{A}$. Then, by 4.11, $\mathcal{L}_{\mathcal{A}}$ is finitary if and only if $\mathcal{A}$ does not contain a reachable silent cycle. By 4.12 this is the case if and only if there is a WTA $\mathcal{A}'$ such that $\mathcal{A} \equiv \mathcal{A}'$. This is in turn equivalent to the fact that $\mathcal{L}_{\mathcal{A}}$ is recognizable. □

4.15 Lemma. *Let $\mathcal{A}$ be a* wWTA. *Then $\mathcal{L}_{\mathcal{A}}$ fails to be a-quasiregular if and only if either there is some $t \in T$ with $\mathrm{dom}(t) = i$, $\mathrm{lab}(t) = a$ or there exists a silent path* $\mathbf{s}$ *starting in i and ending in a state that is the domain of a transition $t \in T$ with $\mathrm{lab}(t) = a$.* □

In the following we will define several operations on wWTAs and show that they relate naturally to the corresponding operations on weighted tree-languages.

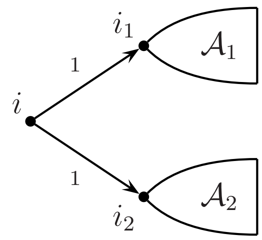
4.16 Product with scalars on wWTAs. Let $\mathcal{A} = (Q, i, T, \lambda, S, \sigma)$ be a wWTA. Then $c \cdot \mathcal{A} := (Q', i', T, \lambda, S', \sigma')$ where $Q' := Q \dot\cup \{i'\}$, $S' := S \dot\cup \{s\}$, $\sigma'(x) := \begin{cases} \sigma(x) & \text{if } x \neq s \\ (i', i, c) & \text{otherwise.} \end{cases}$



4.17 Proposition. $\mathcal{L}_{c \cdot \mathcal{A}} \cong c \cdot \mathcal{L}_{\mathcal{A}}$. □

4.18 Sum of wWTAs. Let $\mathcal{A}_k = (Q_k, i_k, T_k, \lambda_k, S_k, \sigma_k)$, $k \in \{1, 2\}$. Let $i \notin Q_1 \cup Q_2$. Then we define $\mathcal{A}_1 + \mathcal{A}_2 := (Q, i, T, \lambda, S, \sigma)$ according to $Q := Q_1 \dot\cup Q_2 \dot\cup \{i\}$, $T := T_1 \dot\cup T_2$, $\lambda(x) = \begin{cases} \lambda_1(x) & \text{if } x \in T_1 \\ \lambda_2(x) & \text{if } x \in T_2 \end{cases}$, $S := S_1 \dot\cup S_2 \dot\cup \{s_1, s_2\}$,

$$\sigma(x) := \begin{cases} \sigma_1(x) & \text{if } x \in S_1, \\ \sigma_2(x) & \text{if } x \in S_2, \\ (i, i_1, 1) & \text{if } x = s_1, \\ (i, i_2, 1) & \text{if } x = s_2. \end{cases}$$

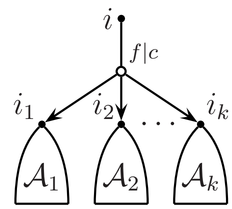


4.19 Proposition. $\mathcal{L}_{\mathcal{A}_1 + \mathcal{A}_2} \cong \mathcal{L}_{\mathcal{A}_1} + \mathcal{L}_{\mathcal{A}_2}$. □

4.20 Topcatenation on wWTAs. Let $f \in \Sigma^{(n)}$, $\mathcal{A}_k = (Q_k, i_k, T_k, \lambda_k, S_k, \sigma_k)$ $(k = 1, \ldots, n)$ be wWTAs and let $c \in K$. Let $i \notin Q_1 \cup \cdots \cup Q_n$. Then we define $[f|c]\langle \mathcal{A}_1, \ldots, \mathcal{A}_n \rangle := (Q, i, T, \lambda, S, \sigma)$ according to: $Q := \{i\} \dot\cup \coprod_{k=1}^n Q_k$, $T := \{\tau\} \dot\cup \coprod_{k=1}^n T_k$,

$$\lambda(x) := \begin{cases} \lambda_k(x) & \text{if } x \in Q_k \quad (k = 1, \ldots, n) \\ (i, f, i_1, \ldots, i_n, c) & \text{if } x = \tau, \end{cases}$$

$S := \coprod_{k=1}^n S_k$ and $\sigma(x) = \sigma_k(x)$ if $x \in S_k$ $(k = 1, \ldots, n)$.

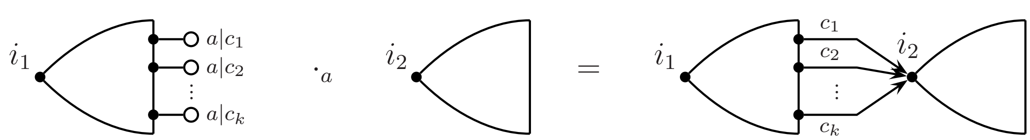


4.21 Proposition. $\mathcal{L}_{[f|c]\langle \mathcal{A}_1, \ldots, \mathcal{A}_n \rangle} = [f|c]\langle \mathcal{L}_{\mathcal{A}_1}, \ldots, \mathcal{L}_{\mathcal{A}_n} \rangle$. □

4.22 a-product of wWTAs. Let $a \in \Sigma^{(0)}$ and let $\mathcal{A}_k = (Q_k, i_k, T_k, \lambda_k, S_k, \sigma_k)$, $k \in \{1, 2\}$ be wWTAs. Let $T_a := \{\tau \in T_1 \mid \lambda(\tau) = (q, a, c),\ q \in Q_1,\ c \in K\}$. We define $\mathcal{A}_1 \cdot_a \mathcal{A}_2 := (Q, i_1, T, \lambda, S, \sigma)$ according to $Q := Q_1 \dot\cup Q_2$, $T := (T_1 \setminus T_a) \dot\cup T_2$,

$$\lambda(x) := \begin{cases} \lambda_1(x) & \text{if } x \in T_1 \setminus T_a \\ \lambda_2(x) & \text{if } x \in T_2, \end{cases},$$

$S := S_1 \dot\cup (s_\tau)_{\tau \in T_a} \dot\cup S_2$ and $\sigma(x) = \begin{cases} \sigma_1(x) & \text{if } x \in S_1, \\ \sigma_2(x) & \text{if } x \in S_2, \\ (q, c, i_2) & \text{if } x = s_\tau,\ \lambda_1(\tau) = (q, a, c), \end{cases}$ where $(s_\tau)_{\tau \in T_a}$ is a family of distinct silent transition-symbols disjoint from S_1 and S_2.



4.23 Proposition. $\mathcal{L}_{\mathcal{A}_1 \cdot_a \mathcal{A}_2} = \mathcal{L}_{\mathcal{A}_1} \cdot_a \mathcal{L}_{\mathcal{A}_2}$.

Proof. We aim at using 2.16. In particular we will construct an isomorphism from the language $(L, |.|)$ to $\mathcal{L}_{\mathcal{A}_1 \cdot_a \mathcal{A}_2}$. For this we introduce the a-substitution of initial runs through $\mathcal{A}_2$ into runs through $\mathcal{A}_1$ the result of which will be a run of $\mathcal{A}_1 \cdot_a \mathcal{A}_2$. While doing this we also prove en passant that a-substitution of runs is compatible with the a-substitution of the corresponding weighted trees. That is

$$|r \circ_a \langle p_1, \dots , p_n\rangle| = |r| \circ_a \langle |p_1|, \dots , |p_n|\rangle. \tag{4}$$

For $\tau \in T_1$ with $\lambda(\tau) = (q, a, c)$ we define $\mathrm{rk}_a(\tau) := 1$. For any initial run p of $\mathcal{A}_2$ the a-substitution $\tau \circ_a \langle p\rangle$ is defined as $s_\tau \cdot p$. Obviously $|\tau \circ_a \langle p\rangle| = |s_\tau \cdot p| = c \cdot |p| = |\tau| \circ_a \langle |p|\rangle$. If $\lambda(\tau) = (q, b, c)$ then we define $\mathrm{rk}_a(\tau) := 0$. In this case $\tau \circ_a \langle\rangle := \tau$.

Let now $r = \tau\langle r_1, \dots , r_n\rangle$ be a run of $\mathcal{A}_1$ with $\lambda(\tau) = (q, f, q_1, \dots , q_n, c)$. Suppose that $\mathrm{rk}_a(r_i) = k_i$ $(i = 1, \dots , n)$. Then we define $\mathrm{rk}_a(r) := \sum_{i=1}^n k_i$ and for initial runs $p_{1,1}, \dots , p_{1,k_1}, \dots , p_{n,k_n}$ of $\mathcal{A}_2$ we define

$$r \circ_a \langle p_{1,1}, \dots , p_{n,k_n}\rangle := \tau\langle r_1 \circ_a \langle p_{1,1}, \dots , p_{1,k_1}\rangle, \dots , r_n \circ_a \langle p_{n,1}, \dots , p_{n,k_n}\rangle\rangle.$$

For showing (4) we note that

$$\begin{aligned}
&|r \circ_a \langle p_{1,1}, \dots , p_{n,k_n}\rangle| \\
&\qquad = |\tau\langle r_1 \circ_a \langle p_{1,1}, \dots , p_{1,k_1}\rangle, \dots , r_n \circ_a \langle p_{n,1}, \dots , p_{n,k_n}\rangle\rangle| \\
&\qquad = [f|c]\langle |r_1 \circ_a \langle p_{1,1}, \dots , p_{1,k_1}\rangle|, \dots , |r_n \circ_a \langle p_{n,1}, \dots , p_{n,k_n}\rangle|\rangle \\
&\qquad = [f|c]\langle |r_1| \circ_a \langle |p_{1,1}|, \dots , |p_{1,k_1}|\rangle, \dots , |r_n| \circ_a \langle |p_{n,1}|, \dots , |p_{n,k_n}|\rangle\rangle \\
&\qquad = [f|c]\langle |r_1|, \dots , |r_n|\rangle \circ_a \langle |p_{1,1}|, \dots , |p_{n,k_n}|\rangle \\
&\qquad = |r| \circ_a \langle |p_{1,1}|, \dots , |p_{n,k_n}|\rangle.
\end{aligned}$$

Let finally $r = s \cdot r'$ be a run through $\mathcal{A}_1$ with $\sigma(s) = (q, c, q')$. Then $\mathrm{rk}_a(r) := \mathrm{rk}_a(r') =: n$ and for initial runs $p_1, \dots , p_n$ through $\mathcal{A}_2$ we define $r \circ_a \langle p_1, \dots , p_n\rangle := s \cdot r' \circ_a \langle p_1, \dots , p_n\rangle$. It remains to show (4):

$$\begin{aligned}
|r \circ_a \langle p_1, \dots , p_n\rangle| &= |s \cdot r' \circ_a \langle p_1, \dots , p_n\rangle| \\
&= c \cdot |r' \circ_a \langle p_1, \dots , p_n\rangle| \\
&= c \cdot |r'| \circ_a \langle |p_1|, \dots , |p_n|\rangle \\
&= |s \cdot r'| \circ_a \langle |p_1|, \dots , |p_n|\rangle \\
&= |r| \circ_a \langle |p_1|, \dots , |p_n|\rangle.
\end{aligned}$$

Now $(L, |.|)$ be defined as in Lemma 2.16. That is

$$L = \{t\langle s_1, \dots , s_n\rangle \mid t \in \mathcal{L}_{\mathcal{A}_1},\, s_1, \dots , s_n \in \mathcal{L}_{\mathcal{A}_2},\, \mathrm{rk}_a(t) = n,\, n \in \mathbb{N}\}$$

and

$$|t\langle s_1, \dots , s_n\rangle| = |t| \circ_a \langle |s_1|, \dots , |s_n|\rangle.$$

We define $\varphi : (L, |.|) \longrightarrow \mathcal{L}_{\mathcal{A}_1 \cdot_a \mathcal{A}_2}$ according to $t\langle s_1, \dots , s_n\rangle \mapsto t \circ_a \langle s_1, \dots , s_n\rangle$. By (4), φ is a homomorphism. In order to prove bijectivity of φ it is important to observe that each run p of $\mathcal{A}_1 \cdot_a \mathcal{A}_2$ that roots in Q_1 has a unique decomposition $r \circ_a \langle p_1, \dots, p_n \rangle$ for some run r of $\mathcal{A}_1$, $\mathrm{rk}_a(r) = n$ (for some $n \in \mathbb{N}$) and initial runs $p_1, \dots, p_n$ of $\mathcal{A}_2$.

First we note that every run of $\mathcal{A}_1 \cdot_a \mathcal{A}_2$ that has its root in Q_2, is essentially a run of $\mathcal{A}_2$. Runs of $\mathcal{A}_1 \cdot_a \mathcal{A}_2$ that start in Q_1 can be described by an induction. The simplest such runs are either of the shape $s_\tau \cdot p_1$ where p_1 is an initial run of $\mathcal{A}_2$ and where $\tau \in T_a$ or they are of the shape τ where $\tau \in T_a^{(0)} \setminus T_a$. The first kind of runs allows a unique decomposition as $\tau \circ_a \langle p_1 \rangle$. The second type has a trivial decomposition $\tau \circ_a \langle \, \rangle$ which is also unique. Thus the induction-base is set.

Suppose now that $p = \tau\langle p_1, \dots, p_n \rangle$ where $\tau \in T_1^{(n)}$, $\lambda(\tau) = (q, f, q_1, \dots, q_n, c)$ for some $n \neq 0$. By induction-hypotheses the p_i have unique decompositions $p_i = r_i \circ_a \langle p_{i,1}, \dots, p_{i,k_i} \rangle$ where the $p_{i,j}$ are initial runs of $\mathcal{A}_2$ and where r_i is a run of $\mathcal{A}_1$ ($i = 1, \dots, n$, $j = 1, \dots, k_i$). With this knowledge we compute

$$\begin{aligned} p &= \tau\langle r_1 \circ_a \langle p_{1,1}, \dots, p_{1,k_1} \rangle, \dots, r_n \circ_a \langle p_{n,1}, \dots, p_{n,k_n} \rangle\rangle \\ &= \tau\langle r_1, \dots, r_n \rangle \circ_a \langle p_{1,1}, \dots, p_{n,k_n} \rangle. \end{aligned}$$

This is the only decomposition of p because the decompositions of the p_i are unique.

Suppose that $p = s \cdot p_1$ for some $s \in S_1$ with $\sigma(s) = (q, c, q')$, then $q' \in Q_1$ and by induction-hypothesis p_1 has a unique decomposition $p_1 = r_1 \circ_a \langle p_{1,1}, \dots, p_{1,k} \rangle$. But then

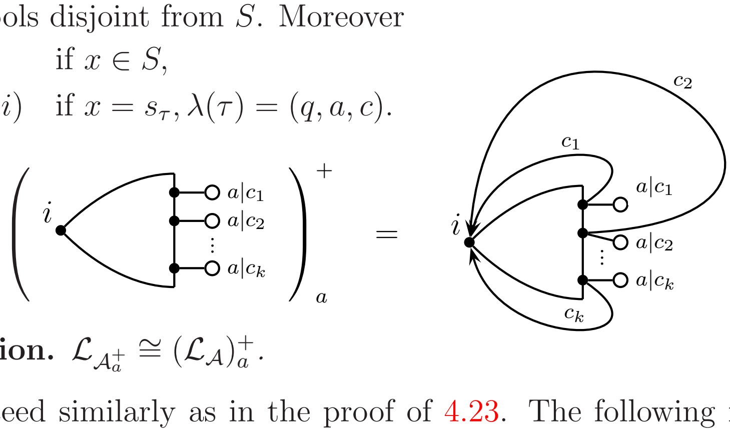
$$\begin{aligned} p &= s \cdot (r_1 \circ_a \langle p_{1,1}, \dots, p_{1,k} \rangle) \\ &= (s \cdot r_1) \circ_a \langle p_{1,1}, \dots, p_{1,k} \rangle \end{aligned}$$

is the unique decomposition of p. This finishes the proof that all runs of $\mathcal{A}_1 \cdot_a \mathcal{A}_2$ rooting in Q_1 are uniquely decomposable.

Altogether $\mathcal{L}_{\mathcal{A}_1 \cdot_a \mathcal{A}_2} \cong (L, |.|) \cong \mathcal{L}_{\mathcal{A}_1} \cdot_a \mathcal{L}_{\mathcal{A}_2}$. □

4.24 a-Semi-iteration of wWTAs. Let $a \in \Sigma^{(0)}$ and let $\mathcal{A} = (Q, i, T, \lambda, S, \sigma)$ be a wWTA. Let $T_a := \{\tau \in T \mid \lambda(\tau) = (q, a, c),\ q \in Q,\ c \in K\}$. We define $\mathcal{A}_a^+ := (Q, i, T, \lambda, S', \sigma')$ where $S' := S \dot\cup (s_\tau)_{\tau \in T_a}$,where $(s_\tau)_{\tau \in T_a}$ is a family of distinct silent transition symbols disjoint from S. Moreover

$$\sigma'(x) := \begin{cases} \sigma(x) & \text{if } x \in S, \\ (q, c, i) & \text{if } x = s_\tau, \lambda(\tau) = (q, a, c). \end{cases}$$



4.25 Proposition. $\mathcal{L}_{\mathcal{A}_a^+} \cong (\mathcal{L}_{\mathcal{A}})_a^+$.

Proof. We proceed similarly as in the proof of 4.23. The following is a sketch of the proof:

1. To each run r of $\mathcal{A}_a^+$ a rank $\mathrm{rk}(r)$ will be assigned.
2. We consider the set $\mathcal{R}$ which contains all runs of $\mathcal{A}_a^+$ and an additional element ε of rank 1. With $\mathcal{R}' := \mathrm{run}(\mathcal{A}_a^+) \cup \{\varepsilon\}$ we define a composition operation $\circ$ of elements from $\mathcal{R}$ with elements from $\mathcal{R}'$. We show that $(\mathcal{R}', \mathrm{rk}, \circ, \varepsilon)$ forms a ranked monoid.
3. We show that $|.| : (\mathcal{R}', \mathrm{rk}, \circ, \varepsilon) \longrightarrow (\mathrm{WT}_\Sigma, \mathrm{rk}_a, \circ_a, [a|1])$ is a homomorphism of ranked monoids (where $|\varepsilon| := [a|1]$).
4. We show that $(\mathcal{R}', \mathrm{rk}, \circ, \varepsilon) \cong (\mathrm{run}(\mathcal{A}), \mathrm{rk}_a)^*$. From this we may conclude then using 2.23 and 2.29, that $(\mathrm{run}(\mathcal{A}_a^+), \mathrm{rk}, \circ) \cong (\mathrm{run}(\mathcal{A}), \mathrm{rk}_a)^+$.

Steps 1–3 will be carried out simultaneously in an induction on the structure of the runs of $\mathcal{A}_a^+$.

We start by setting $\mathrm{rk}(\varepsilon) := 1$, $|\varepsilon| := [a|1]$ and by defining $\varepsilon \circ \langle p \rangle := p$ for all $p \in \mathcal{R}'$. Obviously

$$\mathrm{rk}(\varepsilon) = 1 = \mathrm{rk}_a([a|1]) = \mathrm{rk}_a(|\varepsilon|),$$
$$|\varepsilon \circ \langle p \rangle| = |p| = [a|1] \circ \langle |p| \rangle = |\varepsilon| \circ \langle |p| \rangle.$$

Now for $\tau \in T$ with $\lambda(\tau) = (q, a, c)$ we define $\mathrm{rk}(\tau) := 1$ and we define $\tau \circ \langle \varepsilon \rangle := \tau$ and $\tau \circ \langle p \rangle := s_\tau \cdot p$ for $p \in \mathrm{run}(\mathcal{A}_1^+)$. Obviously $\mathrm{rk}(\tau) = \mathrm{rk}_a(|\tau|)$ (cf. 2.22 and 1.11) and

$$|\tau \circ \langle \varepsilon \rangle| = |\tau| = |\tau| \circ_a \langle [a|1] \rangle = |\tau| \circ_a \langle |\varepsilon| \rangle \quad \text{and}$$
$$|\tau \circ \langle p \rangle| = |s_\tau \cdot p| = c \cdot |p| = |\tau| \circ_a \langle |p| \rangle$$

If on the other hand $\lambda(\tau) = (q, b, c)$ for some $b \neq a$ from $\Sigma^{(0)}$, then we define $\mathrm{rk}(\tau) := 0$. Again it is obvious that $\mathrm{rk}(\tau) = \mathrm{rk}_a(|\tau|)$ (cf. 2.22 and 1.11) and that $|\tau \circ \langle\rangle| = |\tau| = |\tau| \circ_a \langle\rangle$.

Let now $r = \tau\langle r_1, \ldots, r_n \rangle$ be a run of $\mathcal{A}_a^+$ where $\lambda(\tau) = (q, f, q_1, \ldots, q_n, c)$. Assume that $\mathrm{rk}(r_i) = k_i$ $(i = 1, \ldots, n)$. Then $\mathrm{rk}(r) := \sum_{i=1}^n k_i$ (note that $\mathrm{rk}(r) = \sum k_i = \sum \mathrm{rk}_a(|r_i|) = \mathrm{rk}_a(|r|)$). For elements $p_{1,1}, \ldots, p_{1,k_1}, \ldots, p_{n,k_n}$ of $\mathcal{R}'$ we define

$$r \circ \langle p_{1,1}, \ldots, p_{n,k_n} \rangle := \tau\langle r_1 \circ \langle p_{1,1}, \ldots, p_{1,k_1} \rangle, \ldots, r_n \circ \langle p_{n,1}, \ldots, p_{n,k_n} \rangle \rangle.$$

We compute that

$$\begin{aligned}
&|r \circ \langle p_{1,1}, \ldots, p_{n,k_n} \rangle| \\
&\qquad = |\tau\langle r_1 \circ \langle p_{1,1}, \ldots, p_{1,k_1} \rangle, \ldots, r_n \circ \langle p_{n,1}, \ldots, p_{n,k_n} \rangle \rangle| \\
&\qquad = [f|c]\langle |r_1 \circ \langle p_{1,1}, \ldots, p_{1,k_1} \rangle|, \ldots, |r_n \circ \langle p_{n,1}, \ldots, p_{n,k_n} \rangle| \rangle \\
&\qquad = [f|c]\langle |r_1| \circ_a \langle |p_{1,1}|, \ldots, |p_{1,k_1}| \rangle, \ldots, |r_n| \circ_a \langle |p_{n,1}|, \ldots, |p_{n,k_n}| \rangle \rangle \\
&\qquad = [f|c]\langle |r_1|, \ldots, |r_n| \rangle \circ_a \langle |p_{1,1}|, \ldots, |p_{n,k_n}| \rangle \\
&\qquad = |r| \circ_a \langle |p_{1,1}|, \ldots, |p_{n,k_n}| \rangle.
\end{aligned}$$

If, finally, $r = s \cdot r'$ is a run of $\mathcal{A}_a^+$ with $\sigma'(s) = (q_1, q_2, c)$, then $\mathrm{rk}(r) := \mathrm{rk}(r') = \mathrm{rk}_a(|r'|) = \mathrm{rk}_a(|r|)$. Moreover $r \circ \langle p_1, \ldots, p_n\rangle := (s \cdot r') \circ \langle p_1, \ldots, p_n\rangle$. We note that

$$\begin{aligned}|r \circ \langle p_1. \ldots, p_n\rangle| &= |(s \cdot r') \circ \langle p_1, \ldots, p_n\rangle| = c \cdot |r' \circ \langle p_1, \ldots, p_n\rangle| \\ &= (c \cdot |r'|) \circ_a \langle |p_1|, \ldots, |p_n|\rangle = |r| \circ_a \langle |p_1|, \ldots, |p_n|\rangle.\end{aligned}$$

Now we equip $\mathcal{R}'$ with the operation $\circ$ and with the unit ε. This turns it into a ranked monoid. To show the superassociativity of $\circ$ is technical but very simple.

This finishes steps 1-3 of our agenda. Let us turn our attention to step 4 now. By construction of $\mathcal{A}_a^+$, each run of $\mathcal{A}$ is also a run of $\mathcal{A}_a^+$. Hence, the canonical embedding ι of $\mathrm{run}(\mathcal{A})$ into $\mathcal{R}'$ extends uniquely to a homomorphism $\iota^\#$ from the free ranked monoid $(\mathrm{run}(\mathcal{A}), \mathrm{rk}_a)^*$ to $(\mathcal{R}', \mathrm{rk}, \circ, \varepsilon)$. Observe now that the following diagram commutes:

$$\begin{tikzcd}
(\mathcal{R}', \mathrm{rk}, \circ, \varepsilon) \arrow[r, "|.|"] & (WT_\Sigma, \mathrm{rk}_a, \circ_a, [a|1]) \\
(\mathrm{run}(\mathcal{A})^*, \mathrm{rk}_a^*, \circ, \varepsilon) \arrow[u, "\iota^\#"] \arrow[ur, "|.|_a^*"'] &
\end{tikzcd}$$

This follows directly from point (3) above and from the definition of $|.|_a^*$. Hence $\iota^\#$ is a homomorphism from $(\mathcal{L}_\mathcal{A})_a^*$ to $(\mathcal{R}', |.|)$.

In order to show the bijectivity of this homomorphism, it suffices to show that each run of $\mathcal{A}_a^+$ decomposes uniquely into runs of $\mathcal{A}$ with respect to $\circ$. In other words this means that every run of $\mathcal{A}_a^+$ has precisely one preimage in $(\mathrm{run}(\mathcal{A}), \mathrm{rk}_a)^+$. As usual this is done via structural induction. However, for technical reasons we make a little detour. Let $r \neq \varepsilon$ be an element of $\mathcal{R}$. Any expression of the form

$$r = p \circ \langle p_1, \ldots, p_n\rangle \quad \text{for } p \text{ a run of } \mathcal{A}, , p_1, \ldots, p_n \in \mathcal{R}'$$

is called *reduction* of r. If $p = r$ and $p_1 = \cdots = p_n = \varepsilon$ then the decomposition is called trivial. We call r irreducible if it only admits the trivial reduction. Otherwise it is called *reducible*. Obviously the runs of $\mathcal{A}$ are precisely the irreducible elements of $\mathcal{R}$ since the operation $\circ$ always makes its result contain one of the silent transition-symbols $(s_\tau)_{\tau \in T_a}$, but these are not transitions of $\mathcal{A}$. We will show that every reducible run admits precisely one nontrivial reduction.

Then we argue that every reduction decomposes a run into a run of $\mathcal{A}$ and several shorter runs of $\mathcal{A}_a^+$. Thus, applying reduction recursively leads ultimately to a decomposition of the run into runs from $\mathcal{A}$. On the other hand every decomposition into runs of $\mathcal{A}$ induces a reduction of the run. Hence the uniqueness of the reduction also forces the uniqueness of the decomposition.

If $r = \tau$ for $\tau \in T$ then r is a run of $\mathcal{A}$ and is therefore irreducible. Hence it is also uniquely (trivially) decomposable.

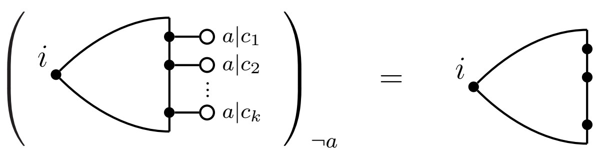
Suppose that $r = \tau\langle p_1, \ldots, p_n\rangle$ where $\tau \in T$ such that each p_i $(i = 1, \ldots, n)$ is either irreducible and hence a run r_i of $\mathcal{A}$, or it is uniquely reducible into $s_i \circ \langle p_{i,1}, \ldots, p_{i,k_i}\rangle$ for a unique run s_i of $\mathcal{A}$ and runs $p_{i,1}, \ldots, p_{i,k_i}$ $(i = 1, \ldots, n)$

each of which is uniquely decomposable. Hence r decomposes as $\tau\langle r_1, \dots, r_n\rangle \circ \langle p_{1,1}, \dots, p_{n,k_n}\rangle$. Since the r_i are unique, this decomposition is also unique.

Consider finally the case where $r = s \cdot p$ with $s \in S'$. If $s \in S$ and if p is reducible to $p' \circ \langle p_1, \dots, p_n\rangle$ for a run p' of $\mathcal{A}$, then r is reducible to $(s \cdot p') \circ \langle p_1, \dots, p_n\rangle$. Since p' is unique, the reduction of r is also unique. If on the other hand $s = s_\tau$ for some $\tau \in T_a$, then $r = \tau \circ \langle p\rangle$ is the unique reduction of r.

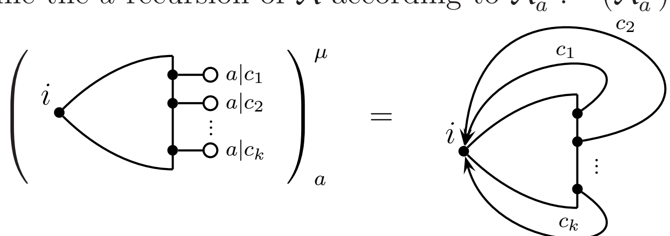
This finishes the proof of $(\mathrm{run}(\mathcal{A}), \mathrm{rk}_a)^+ \cong (\mathrm{run}(\mathcal{A}_a^+), \circ)$. Therefore also the induced homomorphism from $(\mathcal{L}_\mathcal{A})_a^+$ to $\mathcal{L}_{\mathcal{A}_a^+}$ is bijective and is hence an isomorphism. $\square$

4.26 a-Annihilation on wWTAs. Let $a \in \Sigma^{(0)}$ and let $\mathcal{A} = (Q, i, T, \lambda, S, \sigma)$ be a wWTA. Let $T_a := \{\tau \in T \mid \lambda(\tau) = (q, a, c),\ q \in Q,\ c \in K\}$. We define $\mathcal{A}_{\neg a} := (Q, i, T', \lambda', S, \sigma)$ according to $T' := T \setminus T_a$, $\lambda' := \lambda_{|T'}$



4.27 Proposition. $\mathcal{L}_{A_{\neg a}} = (\mathcal{L}_\mathcal{A})_{\neg a}$ $\square$

4.28 a-Recursion of wWTAs. Let $a \in \Sigma^{(0)}$ and let $\mathcal{A} = (Q, i, T, \lambda, S, \sigma)$ be a wWTA. We define the a-recursion of $\mathcal{A}$ according to $\mathcal{A}_a^\mu := (\mathcal{A}_a^+)_{\neg a}$.



4.29 Proposition. $\mathcal{L}_{\mathcal{A}_a^\mu} \cong (\mathcal{L}_\mathcal{A})_a^\mu$.

Proof. First observe that $\mathcal{A}_a^\mu \equiv (\mathcal{A}_a^+)_{\neg a}$. Hence, by 4.27, 4.23 and 2.30, we have

$$\mathcal{L}_{\mathcal{A}_a^\mu} \cong \mathcal{L}_{(\mathcal{A}_a^+)_{\neg a}} \cong (\mathcal{L}_{\mathcal{A}_a^+})_{\neg a} \cong ((\mathcal{L}_\mathcal{A})_a^+)_{\neg a} \cong (\mathcal{L}_\mathcal{A})_a^\mu.$$

$\square$

4.30 Remark. Let $\mathcal{L} \in \mathsf{WTL}_\Sigma$ be weakly recognizable. Then, by Lemma 2.29, $\mathcal{L}_a^* \cong \mathcal{L}_a^+ + \{[a|1]\}$ and since $\{[a|1]\}$ is trivially recognizable we conclude by Propositions 4.19 and 4.25 that $\mathcal{L}_a^*$ is weakly recognizable as well.

4.31 Corollary. *Let $\mathcal{L}$ be a recognizable weighted tree-language and let $a \in \Sigma^{(0)}$. Then $\mathcal{L}_a^*$ is recognizable if and only if $\mathcal{L}$ is a-quasiregular.*

Proof. $\mathcal{L}_a^*$ is weakly recognizable by 4.30. By 4.14 $\mathcal{L}_a^*$ is recognizable if and only if it is finitary. By 2.34 this is the case if and only if $\mathcal{L}$ is a-quasiregular. $\square$

5 Fixed Point Expressions

This section is the core of the thesis. Almost all other results build upon the results obtained here. We start by the definition of fixed point expressions (or for short fp-expressions) together with their wWTA- and WTL-semantics. Each fp-expression defines a weakly recognizable weighted tree-language. Our first main result is that *every* weakly recognizable weighted tree-language is definable by an fp-expression. By a slight restriction of the fp-expressions we obtain the second main result—the characterization of recognizable weighted tree-languages by proper fp-expressions.

5.1 Rules, expressions. In this section and also later on we will define several classes of formal expressions. An expression is a formal word over some alphabet (which is usually given implicitly). Each class C of expressions will be defined inductively as the smallest set of expressions that obeys some given set of rules. These rules will be denoted as follows

$$\text{Label}\ \frac{e_1, \dots, e_n}{e'}\ \text{cond}(e_1, \dots, e_n)$$

where Label denotes the name of the rule, $\text{cond}(e_1, \dots, e_n)$ is a condition on $e_1, \dots, e_n$ and meaning that if $e_1, \dots, e_n$ are expressions from C and if the condition $\text{cond}(e_1, \dots, e_n)$ is fulfilled, then e' is also in C.

In this section we will only deal with a specific class of expressions—the *fixed point expressions*:

5.2 Fixed point expressions. Let $X = (x_i)_{i\in\mathbb{N}}$ be a family of distinct (0-ary) variable symbols disjoint from Σ and let K be a semiring. The set $\text{Fpx}(\Sigma, K)$ of *fixed point expressions* (or briefly, *fp-expressions*) over Σ and K is defined inductively to be the smallest set of expressions that is closed with respect to the following rules:

$$\text{Const}_a\ \frac{}{a} \qquad\qquad \text{Var}_x\ \frac{}{x}$$

$$\text{Top}_f\ \frac{e_1, e_2 \dots e_n}{f\langle e_1, \dots, e_n\rangle}$$

$$\text{Scal}_c\ \frac{e}{c \cdot e} \qquad\qquad \text{Sum}\ \frac{e_1, e_2}{e_1 + e_2}$$

$$\text{Mu}_x\ \frac{e}{\mu x.(e)}$$

where $a \in \Sigma^{(0)}$, $x \in X$, $f \in \Sigma^{(n)}$ for any $n \in \mathbb{N}$ and where $c \in K$.

5.3 Remark. The similarity of fp-expressions to logical formulae was chosen on purpose. The rule Mu_x acts like a quantification of the variable symbol x. Hence we can define bounded and free occurrence of variable symbols in fp-expressions in the usual way. An fp-expression in which no variable symbol occurs freely, is said to be *closed*.

5.4 wWTA-semantics of fp-expressions. The semantics of fp-expressions may be given in terms of wWTAs over the ranked alphabet $\Sigma(X)$.

$$\begin{aligned}
[\![a]\!] &:= (\{i\}, i, \{\tau\}, \lambda, \emptyset, \emptyset) \text{ where } \lambda(\tau) = (i, a, 1),\\
[\![x]\!] &:= (\{i\}, i, \{\tau\}, \lambda, \emptyset, \emptyset) \text{ where } \lambda(\tau) = (i, x, 1),\\
[\![f\langle e_1, \ldots, e_n\rangle]\!] &:= [f|1]\langle [\![e_1]\!], \ldots, [\![e_n]\!]\rangle,\\
[\![c \cdot e]\!] &:= c \cdot [\![e]\!],\\
[\![e_1 + e_2]\!] &:= [\![e_1]\!] + [\![e_2]\!],\\
[\![\mu x.(e)]\!] &:= [\![e]\!]^{\mu}_{x}.
\end{aligned}$$

5.5 WTL-semantics of fp-expressions. The wWTA-semantics of fp-expressions may be used to define a WTL-semantics according to $[\![e]\!]_{\mathsf{WTL}} := \mathcal{L}_{[\![e]\!]}$. In particular, by the results of the previous section, each fp-expression defines a weakly recognizable weighted tree-language in $\mathsf{WTL}_{\Sigma(X)}$. However, many times we like to talk about languages from $\mathsf{WTL}_{\Sigma(X_n)}$. But this is no problem since $\mathsf{WTL}_{\Sigma(X_n)}$ fully embeds into $\mathsf{WTL}_{\Sigma(X)}$. Let $E : \mathsf{WTL}_{\Sigma(X_n)} \longrightarrow \mathsf{WTL}_{\Sigma(X)}$ be the canonical embedding-functor. A weighted tree-language $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X_n)}$ will be called *fp-definable* if there is an fp-expression e whose free variables are from X_n such that $[\![e]\!]_{\mathsf{WTL}} \cong E(\mathcal{L})$. In the sequel we will show that the other direction also holds: each weakly recognizable weighted tree-language is fp-definable.

5.6 Remark. Note, that in [22] the expressions we call “fp-expressions” are called “recognizable tree series expressions” and in [9] the rational expressions are defined using the same iteration operation like in our fp-expressions. However, we chose the name “fp-expressions” since the iteration-rule Mu_x neither generalizes the Kleene-Star for formal languages nor the iteration of formal tree-languages according to [27]. Moreover, the name “recognizable tree series expressions” is inappropriate for us because our primary semantics is not formal tree-series. Moreover the WTL-semantics of an fp-expression is not necessarily recognizable (cf 4.31).

5.7 Proper fixed point expressions. The set $\mathrm{pFpx}(\Sigma, K)$ of *proper fixed point expressions* (or briefly *proper fp-expression*) over Σ and K is defined inductively like in 5.2 where rule Mu_x is replaced by

$$\mathrm{pMu}_x \;\frac{e}{\mu x.(e)}\; [\![e]\!]_{\mathsf{WTL}} \text{ is } x\text{-quasiregular} \qquad (x \in X)$$

5.8 Accessibility graph. In a wWTA transitions connect states. In order to grasp this topological aspect of wWTAs we define the *accessibility graph*. It has the states of the automaton as vertices. An arc is drawn from one state to another whenever there is a transition connecting them. Note that there might be several transitions connecting two states. In this case we also draw multiple arcs between them.

The precise definition goes as follows: Let $\mathcal{A} = (Q, i, T, \lambda, S, \sigma)$ be a wWTA. Let

$$E_1 := \bigcup_{j \in \mathbb{N} \setminus \{0\}} T^{(j)} \times \{1, 2, \ldots, j\},$$
$$E := E_1 \dot{\cup} S.$$

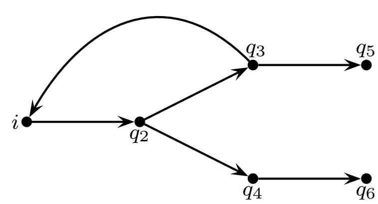
Moreover define

$$s : E \longrightarrow Q \qquad e \mapsto \begin{cases} \mathrm{dom}(t) & e = (t, i),\ t \in T \\ \mathrm{dom}(e) & e \in S \end{cases}$$

$$d : E \longrightarrow Q \qquad e \mapsto \begin{cases} \mathrm{cod}_i(t) & e = (t, i),\ t \in T \\ \mathrm{cod}(e) & e \in S. \end{cases}$$

The multigraph $\Gamma_{\mathcal{A}} = (Q, E, s, d)$ is called *accessibility-graph* of $\mathcal{A}$.[5]

5.9 Example. The accessibility-graph of the wWTA from Example 4.2 is:



5.10 Cyclicity of wWTAs. A *path* of length n in $\Gamma_{\mathcal{A}} = (Q, E, s, d)$ is a word $e_1 e_2 \cdots e_n$ where $e_1, \ldots, e_n \in E$ and such that $d(e_i) = s(e_{i+1})$ $(i = 1, \ldots, n-1)$. Such a path is called *cyclic* if $s(e_1) = d(e_n)$. In particular it is called a *minimal cycle* if for all $1 \leq i < j \leq n$ we have $s(e_i) = s(e_j) \Rightarrow i = j$. The number of minimal cycles of $\Gamma_{\mathcal{A}}$ is called the *cyclicity* of $\mathcal{A}$. It is denoted by $\mathrm{cyc}(\mathcal{A})$.

A state q of $\mathcal{A}$ is called *source* if it is a source of $\Gamma_{\mathcal{A}}$. That is, there does not exist any arc e of $\Gamma_{\mathcal{A}}$ with $d(e) = q$. It is called sink if it is a sink of $\Gamma_{\mathcal{A}}$. That is there is no arc e of $\Gamma_{\mathcal{A}}$ with $s(e) = q$.

[5] The function names s and d are abbreviations for "source" and "destination" of arcs, respectively

5.11 Derivation of wWTAs by transition symbols. Let $\mathcal{A} = (Q, i, T, \lambda, S, \sigma)$ be a wWTA. Let $\tau \in T$ with domain i. Assume $\lambda(\tau) = (i, f, q_1, \ldots, q_n, c)$. For $k \in \{1, \ldots, n\}$ let $Q(q_k)$ be the set of all states reachable by a run in $\mathcal{A}$ starting in q_k,

$$T_k := \{x \in T \mid \mathrm{dom}(x) \in Q(q_k) \setminus \{i\}, x \text{ reachable by a run rooting in } q_k\},$$
$$S_k := \{y \in S \mid \mathrm{dom}(y) \in Q(q_k) \setminus \{i\}, y \text{ reachable by a run rooting in } q_k\},$$
$$\lambda_k := \lambda_{|T_k},$$
$$\sigma_k := \sigma_{|S_k}.$$

The wWTA

$$\frac{\partial \mathcal{A}}{\partial(\tau, k)} := (Q(q_k), q_k, T_k, \lambda_k, S_k, \sigma_k)$$

is called the *derivation of $\mathcal{A}$ by τ at the k-th coordinate*. Moreover we define the *complete derivation* of $\mathcal{A}$ by τ as a tuple:

$$\frac{\partial \mathcal{A}}{\partial \tau} := \left(\frac{\partial \mathcal{A}}{\partial(\tau, 1)}, \ldots, \frac{\partial \mathcal{A}}{\partial(\tau, n)} \right).$$

Let $s \in S$ with $\sigma(s) = (i, q, c)$. Then the *derivation of $\mathcal{A}$ by s* is defined by:

$$\frac{\partial \mathcal{A}}{\partial s} := (Q(q), q, T_s, \lambda_s, S_s, \sigma_s)$$

Where $Q(q)$, T_s, λ_s, S_s and σ_s are defined analogously to $Q(q_k)$, T_k, λ_k, S_k and σ_k above.

In fact, the construction of derivation fixes a new state and reduces the resulting automaton. Thus the derivations do not strictly depend on the respective transition-symbol but only on the new initial state that is in turn an element of the codomain of the transition.

5.12 Remark. Our derivations of automata are somewhat inspired by the Brzozowsky-derivative on rational expressions [10]. Using the Brzozowsky-derivative an automaton can be constructed directly out of a rational expression. Each state of this automaton is then labeled by a rational expression that defines the formal language recognized by the automaton in this state. The initial state is labeled by the original rational expression. Taking the Brzozowsky-derivative of the label of the initial state by a letter a of the alphabet means to follow the (unique) transition with label a from the initial state and to take the codomain of this transition as new initial state of the automaton. The label of the new state is then the Brzozowsky-derivative of the previous expression by a.

5.13 Proposition. *With the notions from above let T_i, S_i be the sets of all transitions with domain i. Then*

$$\mathcal{A} \equiv \sum_{\tau \in T_i} [\mathrm{lab}(\tau) | \, \mathrm{wt}(\tau)] \left\langle \frac{\partial \mathcal{A}}{\partial \tau} \right\rangle \quad + \quad \sum_{s \in S_i} \mathrm{wt}(s) \frac{\partial \mathcal{A}}{\partial s}$$

Proof. We classify the runs of $\mathcal{A}$ by their first transition-symbol. This decomposes $\mathcal{L}_\mathcal{A}$ into a direct sum

$$\coprod_{\tau \in T} \mathcal{L}_\tau + \coprod_{s \in S} \mathcal{L}_s.$$

Then we show that $\mathcal{L}_\tau$ and $\mathcal{L}_s$ are isomorphic to the weighted tree-languages recognized by $[\operatorname{lab}(\tau)|\operatorname{wt}(\tau)]\left\langle \frac{\partial \mathcal{A}}{\partial \tau} \right\rangle$ and $\operatorname{wt}(s) \cdot \frac{\partial \mathcal{A}}{\partial s}$, respectively.

By definition (cf. 4.3) every run p of $\mathcal{A}$ is of either of the following shapes:

1. $p = \tau \qquad (\tau \in T, \operatorname{rk}(\tau) = 0)$,
2. $p = \tau\langle p_1, \dots, p_n \rangle \qquad (\tau \in T)$,
3. $p = s \cdot p' \qquad (s \in S)$.

Case 1: $\mathcal{L}_\tau = \{[\operatorname{lab}(\tau)|\operatorname{wt}(\tau)]\}$. Hence obviously $\mathcal{L}_\tau$ is isomorphic to the weighted tree-language recognized by $[\operatorname{lab}(\tau)|\operatorname{wt}(\tau)]\left\langle \frac{\partial \mathcal{A}}{\partial \tau} \right\rangle$.

Case 2: $\mathcal{L}_\tau$ consists of all runs of $\mathcal{A}$ that start with τ. Let $p \in \mathcal{L}_\tau$. Then $p = \tau\langle p_1, \dots, p_n \rangle$. But then p_i is an initial run through $\frac{\partial \mathcal{A}}{\partial \tau, i}$. On the other hand, in $[\operatorname{lab}(\tau)|\operatorname{wt}(\tau)]\left\langle \frac{\partial \mathcal{A}}{\partial \tau} \right\rangle$ all runs start with a fixed transition-symbol τ' (cf. 4.20). Exchanging in p the initial occurrence of τ by τ' gives the desired isomorphism between $\mathcal{L}_\tau$ and the weighted tree-language recognized by $[\operatorname{lab}(\tau)|\operatorname{wt}(\tau)]\left\langle \frac{\partial \mathcal{A}}{\partial \tau} \right\rangle$.

Case 3: $\mathcal{L}_s$ consists of all runs through $\mathcal{A}$ starting with s. Let $p \in \mathcal{L}_s$, then $p = s \cdot p'$. Clearly p' is a run of $\frac{\partial \mathcal{A}}{s}$. The construction of 4.16 adds precisely one silent transition-symbol s' to $\frac{\partial \mathcal{A}}{\partial s}$. Exchanging in p the first occurrence of s by s' gives the desired isomorphism between $\mathcal{L}_s$ and the weighted tree-language recognized by $\operatorname{wt}(s) \cdot \frac{\partial \mathcal{A}}{\partial s}$. □

5.14 Remark. Note that if i is a source, then the number of states in the derivatives decreases strictly and the cyclicity decreases or remains the same. Hence 5.13 may be used as an induction principle when associating fp-expressions to wWTAs. However, we still need to deal with the case when the initial state is not a source.

5.15 Proposition. *With the notions from above, if $\mathcal{L}_\mathcal{A}$ is finitary then so are $\mathcal{L}_{\frac{\partial \mathcal{A}}{\partial(\tau,k)}}$ $(k = 1, \dots, \operatorname{rk}(\tau))$ and $\mathcal{L}_{\frac{\partial \mathcal{A}}{\partial s}}$.*

Proof. It is obvious that the finite sum of weighted tree-languages is finitary if and only if each summand is finitary. It is similarly clear that the topcatenation of $[f|c]\langle \mathcal{L}_1, \dots, \mathcal{L}_n \rangle$ of weighted tree-languages is finitary if and only if each of the operands $\mathcal{L}_1, \dots, \mathcal{L}_2$ is. Hence, if any of the derivatives defines a non-finitary weighted tree-language then this leads immediately to a contradiction with 5.13. □

5.16 Proposition. *Let $\mathcal{A} = (Q, i, T, \lambda, S, \sigma)$ be a reduced* wWTA *whose initial state i is not a source. Let x be a variable symbol that does not occur as label of any transition-symbol in $\mathcal{A}$. Define $Q' := Q + \{q'\}$ and $T' := T + \{\tau'\}$. For all τ*

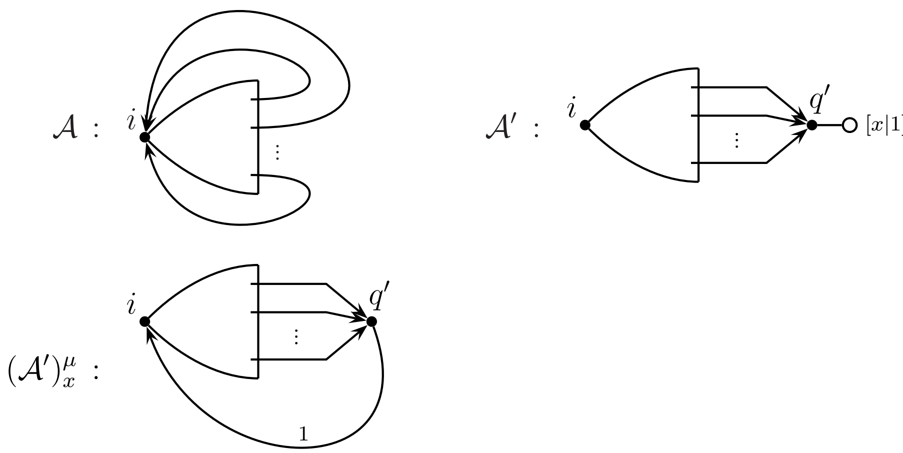
in T *where* $i \notin \mathrm{cod}(\tau)$ *define* $\lambda'(\tau) := \lambda(\tau)$. *For* $\tau \in T$ *with* $i \in \mathrm{cod}(\tau)$ *and with* $\lambda(\tau) = (q, f, q_1, \ldots, q_n, c)$ *we define* $\lambda'(\tau) = (q, f, q'_1, \ldots, q'_n, c)$ *where*

$$q'_k := \begin{cases} q_k & \textit{if } q_k \neq i, \\ q' & \textit{else.} \end{cases}$$

Finally define $\lambda'(\tau') := (q', x, 1)$ *and*

$$\sigma'(s) := \begin{cases} \sigma(s) & \mathrm{cod}(s) \neq i \\ (\mathrm{dom}(s), q', \mathrm{wt}(s)) & \textit{else.} \end{cases}$$

Then the wWTA $\mathcal{A}' = (Q', i, T', \lambda', S, \sigma')$ *is still reduced with* i *being a source. Moreover* $(\mathcal{A}')^{\mu}_{x} \equiv \mathcal{A}$.



Proof. We need to show that $\mathcal{A}$ and $(\mathcal{A}')^{\mu}_{x}$ recognize isomorphic weighted tree languages. To this end we establish a mapping φ from the set of runs of $\mathcal{A}$ to the set of runs of $(\mathcal{A}')^{\mu}_{x}$. The restriction of φ to the initial runs shall then be the desired isomorphism.

In $\mathcal{A}' = (Q', i, T', \lambda', S', \sigma')$ we have $T'_x = \{\tau'\}$. Hence

$$(\mathcal{A}')^{\mu}_{x} = (Q', i, T' \setminus T'_x, \lambda'_{|T' \setminus T'_x}, S'', \sigma'')$$

where $S'' = S' + \{s_{\tau'}\}$ and $\sigma''(s_{\tau'}) = (q', i, 1)$. Note that $T' \setminus T'_x = T$. Hence $(\mathcal{A}')^{\mu}_{x} = (Q', i, T, \lambda'_{|T}, S'', \sigma'')$. Note that $s_{\tau'}$ is the only transition of $(\mathcal{A}')^{\mu}_{x}$ with i in the codomain.

Let us define the mapping φ now. We proceed by induction on the structure of runs r of $\mathcal{A}$.

If $r = \tau$ for $\tau \in T$, then $\varphi(\tau) := \tau$. Obviously $\lambda'(\tau) = \lambda(\tau)$, hence $|\tau| = |\varphi(\tau)|$.

If $r = \tau\langle r_1, \ldots, r_n\rangle$ where $\lambda(\tau) = (q, f, q_1, \ldots, q_n, c)$ and if $i \notin \mathrm{cod}(\tau)$ then we define $\varphi(r) := \tau\langle\varphi(r_1), \ldots, \varphi(r_n)\rangle$. Note that

$$|r| = [f|c]\langle|r_1|, \ldots, |r_n|\rangle = [f|c]\langle|\varphi(r_1)|, \ldots, |\varphi(r_n)|\rangle = |\varphi(r)|.$$

If on the other hand $i \in \mathrm{cod}(\tau)$, then $\varphi(r) := \tau\langle p_1, \dots , p_n\rangle$ where

$$p_i = \begin{cases} s_\tau \cdot \varphi(r_i) & \text{if } \mathrm{dom}(r_i) = i \\ \varphi(r_i) & \text{else.} \end{cases}$$

Here also $|r| = |\varphi(r)|$ since $|s_{\tau'} \cdot \varphi(r_i)| = 1 \cdot |\varphi(r_i)| = |\varphi(r_i)|$.

If $r = s \cdot p$ and if $\mathrm{cod}(s) \neq i$, then $\varphi(r) := s \cdot \varphi(p)$. Otherwise $\varphi(r) := s_{\tau'} \cdot s \cdot \varphi(p)$. Again it is clear that $|r| = |\varphi(r)|$.

We claim now that φ restricted to $\mathrm{run}(\mathcal{A})$ is an isomorphism to from $\mathcal{L}_{\mathcal{A}}$ to $\mathcal{L}_{(\mathcal{A}')^\mu_x}$. We already showed that it is a homomorphism. Now we argue that any run r of $(\mathcal{A}')^\mu_x$ that does not involve $s_{\tau'}$ is already a path of $\mathcal{A}$ and φ leaves r constant. Otherwise deletion of all occurrences of $s_{\tau'}$ from r makes it a run of $\mathcal{A}$ whose image under φ is again r. This shows surjectivity. Since φ only adds the transition $s_{\tau'}$ at several places of a run, injectivity follows as well. □

5.17 Lemma. *With the notions from above* $\mathrm{cyc}(\mathcal{A}') < \mathrm{cyc}(\mathcal{A})$.

Proof. Since all transitions of $\mathcal{A}$ leading to i are redirected in $\mathcal{A}'$ to q', all minimal cycles of $\mathcal{A}$ involving i are destroyed. All other minimal cycles are preserved.

The newly created state q' is a sink. Hence in $\Gamma_{\mathcal{A}'}$ there is no minimal cycle containing q'.

Using that $\mathcal{A}$ is reduced and i is not a source, we conclude that there are minimal cycles through i in $\Gamma_{\mathcal{A}}$. Hence the cyclicity of $\mathcal{A}'$ is strictly smaller than that of $\mathcal{A}$. □

5.18 Proposition. *With the notions from above, if $\mathcal{L}_{\mathcal{A}}$ is finitary then $\mathcal{L}_{\mathcal{A}'}$ is finitary and x-quasiregular.*

Proof. The silent cycles of $\mathcal{A}'$ form a subset of the silent cycles of $\mathcal{A}$. Hence, if $\mathcal{A}$ does not contain a reachable silent cycle then $\mathcal{A}'$ does not contain one either. Consequently, by 4.11, if $\mathcal{L}_{\mathcal{A}}$ is finitary then $\mathcal{L}_{\mathcal{A}'}$ is also finitary. Suppose now that $\mathcal{L}_{\mathcal{A}'}$ is not x-quasiregular. Then by 4.15, it contains a silent path ending in q'. Hence $(\mathcal{A}')^\mu_x$ contains a silent cycle. Since there is an isomorphism between the runs of $\mathcal{A}$ and the runs of $(\mathcal{A}')^\mu_x$, $\mathcal{A}$ contains a silent cycle as well. But all transitions of $\mathcal{A}$ are reachable by assumption, hence the silent cycle is reachable. Therefore, by 4.11, $\mathcal{L}_{\mathcal{A}}$ is not finitary—contradiction. □

5.19 Remark. The constructions from 5.13 and 5.16 give an inductive method for constructing fp-expressions from wWTAs. In the beginning of this process both reductions apply alternatingly—the number of minimal cycles gradually decreases until we reach acyclic wWTAs. In each step either the number of states or the cyclicity decreases. Once we have reached an acyclic automaton the construction from 5.13 applies, gradually decreasing the number of states until we reach a trivial automaton. Hence we finally get:

5.20 Proposition. *Every weakly recognizable weighted tree-language is definable by an fp-expression.*

Proof. We prove by induction that each wWTA recognizes an fp-definable weighted tree-language.

As induction index we associate to each wWTA $\mathcal{A} = (Q, i, T, \lambda, S, \sigma)$ the pair of natural numbers $(\operatorname{cyc}(\mathcal{A}), |Q|)$. On these integer-pairs consider the lexicographical order:

$$(x, y) \leq (u, v) \iff x < u \vee (x = u \wedge y \leq v).$$

Since any wWTA has an initial state, the smallest possible index is $(0, 1)$. Such an automaton has $Q = \{i\}$ and $S = \emptyset$. Moreover if $T = \{t_1, \ldots, t_n\}$ then there are $a_1, \ldots, a_n \in \Sigma^{(0)} \cup X$ and $c_1, \ldots, c_n \in K$ such that $\lambda(t_k) = (i, a_k, c_k)$ $(k = 1, \ldots, n)$. The weighted tree-language that is recognized by such an automaton is $\{[a_1|c_1], \ldots, [a_n|c_n]\}$ this is definable by the following fp-expression:

$$\sum_{k=1}^{n} c_k \cdot a_k. \tag{5}$$

Suppose now the claim holds for all wWTAs with index less than (n, m). Let $\mathcal{A} = (Q, i, T, \lambda, S, \sigma)$ be a wWTA with $\operatorname{cyc}(\mathcal{A}) = n$ and $|Q| = m$.

If i is a source, then we use 5.13 and obtain

$$\mathcal{A} \equiv \sum_{\tau \in T_i} [\operatorname{lab}(\tau) | \operatorname{wt}(\tau)] \left\langle \frac{\partial \mathcal{A}}{\partial \tau} \right\rangle \quad + \quad \sum_{s \in S_i} \operatorname{wt}(s) \frac{\partial \mathcal{A}}{\partial s}$$

For $\tau \in T_i$ of arity k let $\mathcal{A}_{\tau,k} := \frac{\partial \mathcal{A}}{\partial(\tau,k)}$ and for $s \in S_i$ let $\mathcal{A}_s := \frac{\partial \mathcal{A}}{\partial s}$.

Since the number of states of $\mathcal{A}_{\tau,k}$ is strictly smaller than that of $\mathcal{A}$ and the cyclicity of $\mathcal{A}_{\tau,k}$ is not greater that of $\mathcal{A}$, we conclude that the index of $\mathcal{A}_{\tau,k}$ is strictly smaller than that of $\mathcal{A}$. Hence the weighted tree-language that is recognized by $\mathcal{A}_{\tau,k}$ is fp-definable. The same holds for the derivations by silent transitions.

For $j \in \mathbb{N}$, $\tau \in T_i^{(j)}$, $1 \leq k \leq j$ let $e_{\tau,k}$ be an fp-expression defining a weighted tree-language isomorphic to the one recognized by $\mathcal{A}_{\tau,k}$. Moreover, for $s \in S_i$ let e_s be an fp-expression defining a weighted tree-language that is isomorphic to the one recognized by $\mathcal{A}_s$. Then

$$\sum_{j \in \mathbb{N}} \sum_{\tau \in T_i^{(j)}} [\operatorname{lab}(\tau) | \operatorname{wt}(\tau)] \langle e_{t,1}, \ldots, e_{t,j} \rangle \quad + \quad \sum_{s \in S_i} \operatorname{wt}(s) \cdot e_s$$

is an fp-expression defining a weighted tree-language isomorphic to $\mathcal{L}_{\mathcal{A}}$.

If i is not a source then we use 5.16 and obtain a wWTA $\mathcal{A}'$ such that $(\mathcal{A}')^{\mu}_{x} \equiv \mathcal{A}$. With 5.17 we conclude that the index of $\mathcal{A}'$ is strictly smaller than that of $\mathcal{A}$. Hence there is an fp-expression e such that $[\![r]\!] \equiv \mathcal{L}_{\mathcal{A}'}$. Therefore $\mu x.(e)$ is an fp-expression for $\mathcal{L}_{\mathcal{A}}$. □

5.21 Remark. The previous poof corresponds to the inductive procedure in [18, Thm. 4.1]. Though, in contrast to our method, they do not decompose the automaton into simpler automata but instead of this they decompose its semantics—the set of runs of the automaton.

5.22 Theorem. *Let $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X)}$. Then the following are equivalent:*

1. *$\mathcal{L}$ is weakly recognizable,*
2. *$\mathcal{L}$ is definable by an fp-expression.*

Proof. "1 $\Rightarrow$ 2:" By 5.20, if $\mathcal{L}$ is weakly recognizable, then it is definable by an fp-expression.

"2 $\Rightarrow$ 1:" By 5.5 every fp-expression e defines a wWTA $[\![e]\!]$. Hence $[\![e]\!]_{\mathsf{WTL}}$ is weakly recognizable. □

5.23 Theorem. *Let $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X)}$. Then the following are equivalent:*

1. *$\mathcal{L}$ is recognizable,*
2. *$\mathcal{L}$ is definable by a proper fp-expression.*

Proof. This proof will appeal to the notations of the proof of 5.22.

"1 $\Rightarrow$ 2:" If $\mathcal{L}$ is recognizable, then using 5.22 we can construct an fp-expression that defines $\mathcal{L}$. Let $\mathcal{A}$ be an automaton recognizing $\mathcal{L}$. If the index of $\mathcal{A}$ is $(0, 1)$, then the fp-expression (5) corresponding to $\mathcal{A}$ is proper.

Suppose now $\mathcal{A}$ has index (n, m) and that the claim holds for all indices smaller than (n, m).

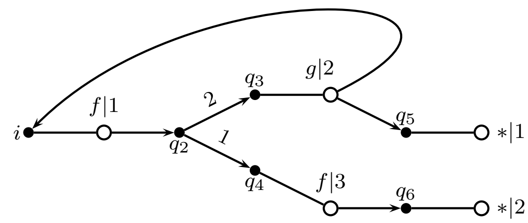
If the initial state of $\mathcal{A}$ is a source, then we we use 5.13 to decompose $\mathcal{A}$ into its derivatives. Since $\mathcal{A}$ contains no reachable silent cycles, its derivatives also do not contain reachable silent cycles. Hence, by 4.11 and by 4.14 their weighted tree-languages are recognizable. However, the new automata have a strictly smaller index. Hence, by induction hypothesis their languages are definable by a proper fp-expression. Since the sum and the topcatenation of proper fp-expressions is proper again, we conclude that $\mathcal{L}$ is definable by a proper fp-expression.

If the initial state of $\mathcal{A}$ is not a source, then 5.16 is used to decompose $\mathcal{A}$. However, this construction does never introduce new cycles but only destroys some of them. Since by 4.14 and by 4.11 $\mathcal{A}$ has no reachable silent cycles, we conclude that the automaton $\mathcal{A}'$ that is constructed in 5.16 has no silent cycles, either. Hence by 4.11 and by 4.14, $\mathcal{A}'$ defines a recognizable weighted tree-language. Since the index of $\mathcal{A}'$ is smaller than the one of $\mathcal{A}$, we conclude, using the induction hypothesis, that $\mathcal{L}_{\mathcal{A}'}$ is definable by a proper fp-expression e. By 5.18 we obtain that $\mathcal{A}'$ is x-quasiregular. Hence $\mu x.(e)$ is a proper fp-expression, that defines $\mathcal{L}$.

"2 $\Rightarrow$ 1:" This follows directly from the fact, that sum and topcatenation and x-annihilation preserve finitarity (cf 2.33). Hence they preserve recognizability (cf. 4.14). Moreover, the x-recursion is only applied to such expressions that define x-quasiregular weighted tree-languages. But on such languages, the x-recursion also preserves recognizability (cf. 4.31 and use 2.27). □

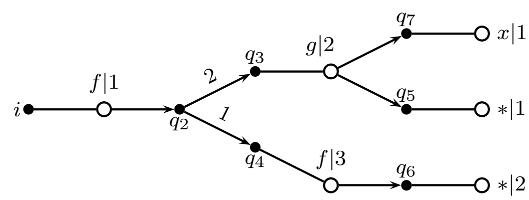
5.24 Example. Let us demonstrate, how to find an fp-expression for a given automaton $\mathcal{A}_0$:

$\mathcal{A}_0$:



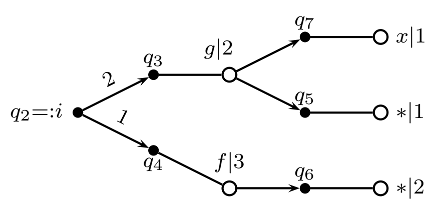
$\mathcal{A}_0 \equiv \mu x.(\mathcal{A}_1)$

$\mathcal{A}_1$:



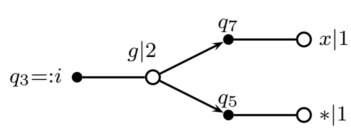
$\mathcal{A}_0 \equiv \mu x.(f\langle \mathcal{A}_2 \rangle)$

$\mathcal{A}_2$:

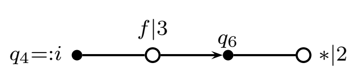


$\mathcal{A}_0 \equiv \mu x.(f\langle 2 \cdot \mathcal{A}_3 + \mathcal{A}_4 \rangle)$

$\mathcal{A}_3$:



$\mathcal{A}_4$:



$\ldots \mathcal{A}_0 \equiv \mu x.(f\langle 2 \cdot 2 \cdot g\langle x, * \rangle + 3 \cdot f\langle 2 \cdot * \rangle \rangle)$

6 Rational Expressions

In this section we will demonstrate several ways to characterize weakly recognizable and recognizable weighted tree-languages as well as recognizable formal power-series by formal expressions. All of these results will base on our findings about fp-expressions from Section 5. In the literature, formal expressions for recognizable formal tree-series and tree-languages are usually called “recognizable tree-series expressions” or “rational expressions”. The latter name is akin to the classical concept of rational expressions for recognizable formal languages. Unfortunately the literature is not consistent in the definition of what is a rational expression in the context of formal tree-series. For instance, Bozapalidis [9] uses as iteration-operation in his rational expressions the operation that we called x-recursion. On the other hand, Droste and Vogler [18] define their rational expressions taking as iteration-operation the x-iteration. The recognizable tree-series expressions by Ésik and Kuich [22] are just our fp-expressions.

We introduced the new name “fp-expression” because in our opinion these expressions do not deserve the name “rational expressions”. The reason is that they do not properly generalize the concept of rational expressions for formal languages and formal power-series. To be more precise, the operation of x-recursion is not a generalization of the Kleene-star (though it is closely related). In the following, when we define rational expressions, we do not like to give preference to any of the available definitions. Instead we will propose several classes of expressions of equal expressive power.

6.1 New rules. Let $X = (x_i)_{i\in\mathbb{N}}$ be a family of variable symbols disjoint from Σ. Let K be a semiring. Subsequently we will extend our collection of rules that we started in 5.2 by several new rules:

$$\mathrm{Prod}_x\ \frac{e_1, e_2}{e_1 \cdot_x e_2}$$

$$\mathrm{Zero}\ \frac{}{\emptyset} \qquad\qquad \mathrm{Neg}_x\ \frac{e}{(e)_{\neg x}}$$

$$\mathrm{Star}_x\ \frac{e}{(e)_x^*} \qquad\qquad \mathrm{Plus}_x\ \frac{e}{(e)_x^+}$$

for all x in X.

6.2 Semantics of new rules. The wWTA semantics for the new constructs are defined in the obvious way:

$$[\![e_1 \cdot_x e_2]\!] := [\![e_1]\!] \cdot_x [\![e_2]\!] \qquad\qquad [\![\emptyset]\!] := (\{i\}, \emptyset, \emptyset, \emptyset, \emptyset)$$

$$[\![(e)_{\neg x}]\!] := [\![e]\!]_{\neg x} \qquad\qquad [\![(e)_x^*]\!] := [\![e]\!]_x^*$$

$$[\![(e)_x^+]\!] := [\![e]\!]_x^+ .$$

The corresponding WTL-semantics are obvious. We will call a weighted tree-language definable in a set of rules $\mathcal{R}$ if it is isomorphic to the WTL-semantics of an element from the class of expressions defined by $\mathcal{R}$.

6.3 Remark. Many times we are working with weighted tree-languages from $\mathsf{WTL}_{\Sigma(X_n)}$ instead of $\mathsf{WTL}_{\Sigma(X)}$ and we would like to define such languages by expressions. This is no problem since we have that $\mathsf{WTL}_{\Sigma(X_n)}$ is fully embeddable (as a category) into $\mathsf{WTL}_{\Sigma(X)}$. If E is the embedding functor, then we say that $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X_n)}$ is definable by a set of rules $\mathcal{R}$ if $E(\mathcal{L})$ is definable in $\mathcal{R}$.

6.4 Theorem. *Let $\mathcal{L} \in \mathit{WTL}_{\Sigma(X)}$. Let*

$$\mathcal{R} := \{\mathrm{Cons}_a, \mathrm{Var}_x, \mathrm{Top}_f, \mathrm{Scal}_c, \mathrm{Sum} \mid a \in \Sigma^{(0)}, x \in X, f \in \Sigma, c \in K\}.$$

Then the following are equivalent:

1. *$\mathcal{L}$ is weakly recognizable,*
2. *$\mathcal{L}$ is definable through*
$$\mathcal{R} \cup \{\mathrm{Zero}, \mathrm{Prod}_x, \mathrm{Star}_x \mid x \in X\}$$
3. *$\mathcal{L}$ is definable through*
$$\mathcal{R} \cup \{\mathrm{Star}_x, \mathrm{Neg}_x \mid x \in X\}$$
4. *$\mathcal{L}$ is definable through*
$$\mathcal{R} \cup \{\mathrm{Zero}, \mathrm{Prod}_x, \mathrm{Plus}_x \mid x \in X\}$$
5. *$\mathcal{L}$ is definable through*
$$\mathcal{R} \cup \{\mathrm{Plus}_x, \mathrm{Neg}_x \mid x \in X\}.$$

Proof. From Theorem 5.22 we know that $\mathcal{L}$ is weakly recognizable if and only if it is definable by

$$\{\mathrm{Cons}_a, \mathrm{Var}_x, \mathrm{Top}_f, \mathrm{Scal}_c, \mathrm{Sum}, \mathrm{Mu}_x \mid a \in \Sigma^{(0)},\ x \in X,\ c \in K\}.$$

From the results of Section 4 we know that none of the rules considered above lead out of the class of weakly recognizable weighted tree-languages. Hence we only need to show that the provided rule sets are strong enough to capture all such languages. To accomplish this it is enough to show that each rule set is able to simulate the rule Mu_x by a sequence of given rules semantically.

About 2: By 2.27 and 2.24 we have $\mathcal{L}_x^\mu \cong (\mathcal{L}_x^*)_{\neg x} \cong \mathcal{L}_x^* \cdot_x \emptyset$ Hence Mu_x may be simulated by

$$\mathrm{Prod}_x \frac{\mathrm{Star}_x \dfrac{e}{(e)_x^*} \qquad \mathrm{Zero} \dfrac{}{\emptyset}}{(e)_x^* \cdot_a \emptyset}\,.$$

About 3: By 2.27, Mu_x may be simulated by

$$\mathrm{Star}_x \frac{e}{\mathrm{Neg}_x \dfrac{(e)_x^*}{((e)_x^*)_{\neg x}}}\,.$$

About 4: By 2.30 and 2.24, Mu_x may be simulated by

$$\mathrm{Prod}_x \frac{\mathrm{Plus}_x \dfrac{e}{(e)_x^+} \qquad \mathrm{Zero} \dfrac{}{\emptyset}}{(e)_x^+ \cdot_a \emptyset}\,.$$

About 5: By 2.30, Mu_x may be simulated by

$$\mathrm{Plus}_x \frac{e}{\mathrm{Neg}_x \dfrac{(e)_x^+}{((e)_x^+)_{\neg x}}}\,.$$

□

6.5 Remark. From the theory of formal languages we know slightly different version of the Kleene-Theorem. It says that the set of recognizable formal languages is the smallest set of formal languages that contains all finite languages and that is closed with respect to union, product and Kleene-star. In the following we give such a characterization of weakly recognizable weighted tree-languages:

6.6 Corollary. *The class of weakly recognizable weighted tree-languages over $\Sigma(X)$ is the smallest class that contains all finite elements of $\mathsf{WTL}_{\Sigma(X)}$ and that is closed with respect to isomorphic copies, sum, x-product ($x \in X$) and either of x-iteration, x-semiiteration and x-recursion ($x \in X$).*

Proof. By 5.22 it is enough to express products with scalars, topcatenation and x-recursion by finite weighted tree-languages and the above mentioned operations.

For products with scalars we readily agree that

$$c \cdot \mathcal{L} \cong \{[x|c]\} \cdot_x \mathcal{L}.$$

For topcatenation we observe that

$$[f|c]\langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle \cong \{[f|c]\langle [y_1|1], \dots, [y_n|1]\rangle\} \cdot_{y_1} \mathcal{L}_1 \cdots \cdot_{y_n} \mathcal{L}_n$$

where $y_1, \dots, y_n$ are elements of X that occur in neither element of $\mathcal{L}_1, \dots, \mathcal{L}_n$. Such variables exist since $\mathcal{L}_1, \dots, \mathcal{L}_n$ may be assumed to be weakly recognizable and wWTAs contain just a finite number of 0-ary transition-symbols.

About the x-recursion we note that by 2.27 and 2.24 we have

$$\mathcal{L}_x^\mu \cong (\mathcal{L}_x^*)_{\neg x} \cong \mathcal{L}_x^* \cdot_x \emptyset$$

and by 2.30 and 2.24 we have

$$\mathcal{L}_x^\mu \cong (\mathcal{L}_x^+)_{\neg x} \cong \mathcal{L}_x^+ \cdot_x \emptyset.$$

This finishes the proof. □

6.7 Remarks. Having characterized the weakly recognizable weighted tree-languages in so many ways we would like to extend our result from 5.23 to other classes of expressions. Previously we were successful by replacing the rule Mu_x by pMu_x. It comes to no surprise that the same idea will carry over the results from 6.4 to recognizable weighted tree-languages.

6.8. Let us extend our rule-set by proper iteration rules. Our assumptions are the same as in 6.1:

$$\mathrm{pStar}_x\ \frac{e}{(e)_x^*}\ [\![e]\!] \text{ is } x\text{-quasiregular} \qquad \mathrm{pPlus}_x\ \frac{e}{(e)_x^+}\ [\![e]\!] \text{ is } x\text{-quasiregular} \qquad (x \in X)$$

6.9 Theorem. *Let $\mathcal{L} \in \mathit{WTL}_{\Sigma(X)}$. Let*

$$\mathcal{R} := \{\mathrm{Cons}_a, \mathrm{Var}_a, \mathrm{Top}_f, \mathrm{Scal}_c, \mathrm{Sum} \mid c \in K, x \in X, a \in \Sigma^{(0)}\}.$$

Then the following are equivalent:

1. *$\mathcal{L}$ is recognizable,*
2. *$\mathcal{L}$ is definable through*
$$\mathcal{R} \cup \{\mathrm{Zero}, \mathrm{Prod}_x, \mathrm{pStar}_x \mid x \in X\},$$
3. *$\mathcal{L}$ is definable through*
$$\mathcal{R} \cup \{\mathrm{Neg}_x, \mathrm{pStar}_x \mid x \in X\},$$
4. *$\mathcal{L}$ is definable through*
$$\mathcal{R} \cup \{\mathrm{Zero}, \mathrm{Prod}_x, \mathrm{pPlus}_x \mid x \in X\},$$

5. *$\mathcal{L}$ is definable through*

$$\mathcal{R} \cup \{\mathrm{Neg}_x, \mathrm{pPlus}_x \mid x \in X\}.$$

Proof. From 5.23 we know that $\mathcal{L}$ is recognizable if and only if it is definable by $\mathcal{R} \cup \{\mathrm{pMu}_x \mid x \in X\}$. From 2.33 we know that the operations of sum, product with scalars, topcatenation, x-product and x-negation preserve the class of finitary weighted tree-languages. From 4.31 we know that the x-iteration of an x-quasiregular recognizable weighted tree-language is recognizable again. Since obviously the sum of two weighted tree-languages is finitary if and only if the summands are, we conclude by 2.29 that the x-semiiteration of recognizable x-quasiregular weighted tree-languages is also recognizable. Altogether we conclude that the rules from $\mathcal{R}$ together with

$$\{\mathrm{Zero}, \mathrm{Prod}_x, \mathrm{pStar}_x, \mathrm{pPlus}_x, \mathrm{Neg}_x \mid x \in X\}$$

only produce recognizable weighted tree-languages.

By 5.23 it remains to show that each of the above given rule-sets is able to simulate pMu_x semantically. But this is done in the same way as in the proof of 6.4.

About 2: pMu may be simulated by

$$\mathrm{Prod}_x \frac{\mathrm{pStar}_x \dfrac{e}{(e)_x^*} \qquad \mathrm{Zero} \dfrac{}{\emptyset}}{(e)_x^* \cdot_a \emptyset}.$$

About 3: pMu may be simulated by

$$\mathrm{pStar}_x \frac{e}{\mathrm{Neg}_x \dfrac{(e)_x^*}{((e)_x^*)_{\neg x}}}.$$

About 4: pMu may be simulated by

$$\mathrm{Prod}_x \frac{\mathrm{pPlus}_x \dfrac{e}{(e)_x^+} \qquad \mathrm{Zero} \dfrac{}{\emptyset}}{(e)_x^+ \cdot_a \emptyset}.$$

About 5: pMu may be simulated by

$$\mathrm{pPlus}_x \frac{e}{\mathrm{Neg}_x \dfrac{(e)_x^+}{((e)_x^+)_{\neg x}}}.$$

□

6.10 Corollary. *The class of all recognizable weighted tree-languages over $\Sigma(X)$ is the smallest subclass of $\mathsf{WTL}_{\Sigma(X)}$ that contains all finite elements of $\mathsf{WTL}_{\Sigma(X)}$ and that is closed with respect to isomorphic copies, sum, x-product ($x \in X$) and either of x-iteration, x-semiiteration and x-recursion where each of the three iteration operations is restricted to x-quasiregular weighted tree-languages ($x \in X$).*

Proof. Analogous to the proof of 6.6. □

6.11 (weak) rational closure. At the end of this section let us introduce another convenient notion from formal language-theory for weighted tree-languages – the rational closure. Given any class $C \subseteq \mathsf{WTL}_{\Sigma(X)}$ we define the *weak rational closure* $\mathrm{wRat}(C)$ as the smallest class that contains C and $\{\{[x|1]\} \mid x \in X\}$ and that is closed with respect to isomorphic copies, sum, product with scalars, x-product and x-recursion ($x \in X$). The *rational closure* $\mathrm{Rat}(C)$ be the smallest subclass of $\mathsf{WTL}_{\Sigma(X)}$ that contains C and $\{\{[x|1]\} \mid x \in X\}$ and that is closed with respect to x-product and x-recursion ($x \in X$) where x-recursion is restricted to x-quasiregular languages.

Weak rational closures and rational closures are only going to play a role much later in Section 9 about fixed point theory on weighted tree-languages (cf 9.16).

7 Formal Tree-Series

In this section we are finally going to consider formal tree-series and rational operations on them. After the basic definitions we show how each finitary weighted tree-language corresponds to a formal tree-series. We introduce the notions of recognizability and a-quasiregularity on formal tree-series and note that they are compatible with the respective notions for weighted tree-languages. Then we take some time to introduce the operations that we already had defined for weighted tree-languages, on formal tree-series and show in each case that they are compatible under the correspondence between weighted tree-languages and formal tree-series. Having this information in hand we go on and translate the Kleene-type results for weighted tree-languages to formal tree-series.

7.1 Formal tree-series. Let (Σ, rk) be a ranked alphabet and let T_Σ be the set of all trees over Σ (cf. 1.1). Let $(K, \oplus, \odot, 0, 1)$ be a semiring. A function $S : \mathrm{T}_\Sigma \longrightarrow K$ is called *formal tree-series*. We will adopt the usual notation for formal tree-series:

$$S = \sum_{t \in \mathrm{T}_\Sigma} (S, t) \cdot t$$

meaning that $S : t \mapsto (S, t) \in K$. Note that the sum is just formal. In particular it has nothing to do with the operations of the semiring. With FTS_Σ we will denote the set of all formal tree-series over Σ.

7.2 Support, Polynomials. The *support* of a formal tree-series S is the set $\mathrm{supp}(S)$ of trees $t \in \mathrm{T}_\Sigma$ for which $(S, t) \neq 0$. If $\mathrm{supp}(S)$ is finite then we call S a *polynomial*. In particular, if $\mathrm{supp}(S) = \{t_1, \ldots, t_n\}$ then we write

$$S = (S, t_1)t_1 + (S, t_2)t_2 + \cdots + (s, t_n)t_n.$$

If a coefficient is equal to 1 then we omit it in the formula for S.

7.3 Weight-function on weighted trees. Let WT_Σ be the set of all weighted trees over Σ with weights from K (cf. 1.3). To each weighted tree t we associate its weight $\mathrm{wt}(t) \in K$. It is defined inductively: For $a \in \Sigma^{(0)}$, $c \in K$ we set $\mathrm{wt}([a|c]) := c$. For $f \in \Sigma^{(n)}$ and $t_1, \ldots, t_n \in \mathrm{WT}_\Sigma$ and $c \in K$ we define $\mathrm{wt}([f|c]\langle t_1, \ldots, t_n\rangle) := c \odot \bigodot_{i=1}^{n} \mathrm{wt}(t_i)$.

7.4 Lemma. *For $c \in K$ and $t \in \mathrm{WT}_\Sigma$ we have*

$$\mathrm{wt}(c \cdot t) = c \odot \mathrm{wt}(t).$$

Assume, K is commutative. Then for $t \in \mathrm{WT}_\Sigma$ with $\mathrm{rk}_a(t) = n$ and for $s_1, \ldots, s_n \in \mathrm{WT}_\Sigma$ we have

$$\mathrm{wt}(t \circ_a \langle s_1, \ldots, s_n \rangle) = \mathrm{wt}(t) \odot \bigodot_{i=1}^{n} \mathrm{wt}(s_i).$$

Proof. As usual we proceed by induction on the structure of t:
about the first claim:

$$\mathrm{wt}(c \cdot [a|d]) = \mathrm{wt}([a|c \odot d]) = c \odot d = c \odot \mathrm{wt}([a|d]).$$

$$\begin{aligned}\mathrm{wt}(c \cdot [f|d]\langle t_1, \ldots, t_n \rangle) &= \mathrm{wt}([f|c \odot d]\langle t_1, \ldots, t_n \rangle) \\ &= c \odot d \odot \bigodot_{i=1}^{n} \mathrm{wt}(t_i) \\ &= c \odot \mathrm{wt}([f|d]\langle t_1, \ldots, t_n \rangle).\end{aligned}$$

about the second claim:

$$\begin{aligned}&\mathrm{wt}([a|c] \circ_a \langle t \rangle) = \mathrm{wt}(c \cdot t) = c \odot \mathrm{wt}(t) = \mathrm{wt}([a|c]) \odot \mathrm{wt}(t) \\ &\mathrm{wt}([b|c] \circ_a \langle\rangle) = \mathrm{wt}([b|c]).\end{aligned}$$

$$\begin{aligned}&\mathrm{wt}([f|c]\langle t_1, \ldots, t_n \rangle \circ_a \langle s_{1,1}, \ldots, s_{n,m_n} \rangle) \\ &\quad = \mathrm{wt}([f|c]\langle t_1 \circ_a \langle s_{1,1}, \ldots, s_{1,m_2} \rangle, \ldots, t_n \circ_a \langle s_{n,1}, \ldots, s_{n,m_n} \rangle \rangle) \\ &\quad = c \odot \bigodot_{i=1}^{n} \mathrm{wt}(t_1 \circ_a \langle s_{i,1}, \ldots, s_{i,m_i} \rangle) \\ &\quad = c \odot \bigodot_{i=1}^{n} \left(\mathrm{wt}(t_i) \odot \bigodot_{j=1}^{m_i} \mathrm{wt}(s_{i,j}) \right) \qquad \text{(by hypothesis)} \\ &\quad = \left(c \odot \bigodot_{i=1}^{n} \mathrm{wt}(t_i) \right) \odot \left(\bigodot_{i=1}^{n} \bigodot_{j=1}^{m_i} \mathrm{wt}(s_{i,j}) \right) \quad (K \text{ is commutative}) \\ &\quad = \mathrm{wt}([f|c]\langle t_1, \ldots, t_n \rangle) \odot \bigodot_{i=1}^{n} \bigodot_{j=1}^{m_i} \mathrm{wt}(s_{i,j})\end{aligned}$$

□

7.5 Formal tree-series for weighted tree-languages. Given now a finitary $\mathcal{L} = (L, |.|) \in \mathsf{WTL}_\Sigma$ we associate a formal tree-series $S_\mathcal{L}$ with $\mathcal{L}$ according to:

$$(S_\mathcal{L}, t) := \bigoplus_{\substack{s \in L \\ \mathrm{ut}(|s|) = t}} \mathrm{wt}(|s|) \qquad (t \in \mathrm{T}_\Sigma).$$

Since $\mathcal{L}$ is finitary, $S_\mathcal{L}$ is well-defined.

7.6 Quasiregularity, recognizability. Let $S \in \mathsf{FTS}_\Sigma$ and $a \in \Sigma^{(0)}$. Then S is called

1. *a-quasiregular* if $(S, a) = 0$,
2. *recognizable* if there is a recognizable (and hence finitary) weighted tree-language $\mathcal{L}$ such that $S = S_\mathcal{L}$.

It is easy to see that if a finitary weighted tree-language is a-quasiregular, then so is its associated formal tree series.

7.7 Remarks. At this point we have to ask the question whether our concept of recognizability matches the usual definitions. In the literature a difference is made between bottom-up recognizable and top-down recognizable formal tree-series. The recognizability concept that we obtain depends only on the definition of the function $\mathrm{wt} : \mathrm{WT}_\Sigma \longrightarrow K$. This function multiplies the weights of the input weighted tree. Since K is not generally commutative, the result depends on the order in which we multiply the weights. Here we chose the following order of the factors. If w_1, w_2 are addresses of the weighted tree such that w_1 is lexicographically smaller than w_2, then the weight of w_1 appears left of the weight of w_2 in the multiplication. This choice for wt leads exactly to top-down-recognizable formal tree-series.

In order to be able to translate our results from the previous section to formal tree-series, the two properties of wt stated in 7.4 will be absolutely crucial. For these to hold, K must be commutative. However, with this assumption the order in which we multiply the weights of a weighted tree does not matter any more. In particular the concepts of top-down- and bottom-up-recognizability coincide for commutative semirings (cf. [8]).

7.8 Operations on FTS_Σ. Next we define the operations of sum, product with a scalar, a-product, topcatenation, a-iteration, a-semiiteration and a-recursion of formal tree-series. While doing so we will also point out their close relations with the corresponding operations on weighted tree-languages.

7.9 Sum. Let $S_1, S_2 \in \mathsf{FTS}_\Sigma$. Then we define $S_1 + S_2$ according to

$$(S_1 + S_2, t) := (S_1, t) \oplus (S_2, t).$$

7.10 Lemma. *Let $\mathcal{L}_1, \mathcal{L}_2 \in \mathsf{WTL}_\Sigma$ be finitary. Then*

$$S_{\mathcal{L}_1 + \mathcal{L}_2} = S_{\mathcal{L}_1} + S_{\mathcal{L}_2}.$$

Proof.

$$\begin{aligned}(S_{\mathcal{L}_1+\mathcal{L}_2}, t) &= \bigoplus_{\substack{s\in\mathcal{L}_1+\mathcal{L}_2\\ \mathrm{ut}(|s|)=t}} \mathrm{wt}(|s|)\\ &= \bigoplus_{\substack{s\in\mathcal{L}_1\\ \mathrm{ut}(|s|)=t}} \mathrm{wt}(|s|) \oplus \bigoplus_{\substack{s\in\mathcal{L}_2\\ \mathrm{ut}(|s|)=t}} \mathrm{wt}(|s|)\\ &= (S_{\mathcal{L}_1}, t) \oplus (S_{\mathcal{L}_2}, t).\end{aligned}$$

□

7.11 Product with scalars. Let $S \in \mathsf{FTS}_\Sigma$, $c \in K$. Then we define $c\cdot S \in \mathsf{FTS}_\Sigma$ according to

$$(c \cdot S, t) := c \odot (S, t) \qquad (t \in \mathrm{T}_\Sigma).$$

7.12 Lemma. *Let $\mathcal{L} = (L, |.|) \in \mathsf{WTL}_\Sigma$ be finitary. Then for $c \in K$ we have $S_{c\cdot\mathcal{L}} = c \cdot S_{\mathcal{L}}$.*

Proof. Let $c \cdot \mathcal{L} = (L, |.|_c)$. Then for $s \in L$: $|s|_c = c \cdot |s|$. Hence, by 7.4(1.1), $\mathrm{wt}(|s|_c) = c \cdot \mathrm{wt}(|s|)$. □

7.13 a-Product. Let $S_1, S_2 \in \mathsf{FTS}_\Sigma$ and let $a \in \Sigma^{(0)}$. Then we define $S_1 \cdot_a S_2$ according to

$$(S_1 \cdot_a S_2, t) := \bigoplus_{\substack{s\in\mathrm{T}_\Sigma\\ s_1,\dots,s_{\mathrm{rk}_a(s)}\in\mathrm{T}_\Sigma\\ t=s\circ_a\langle s_1,\dots,s_{\mathrm{rk}_a(s)}\rangle}} (S_1, s) \odot \bigodot_{i=1}^{\mathrm{rk}_a(s)} (S_2, s_i)$$

7.14 Lemma. *Assume, K is commutative. Then for $\mathcal{L}_1, \mathcal{L}_2 \in \mathsf{WTL}_\Sigma$ we have*

$$S_{\mathcal{L}_1\cdot_a\mathcal{L}_2} = S_{\mathcal{L}_1} \cdot_a S_{\mathcal{L}_2}.$$

Proof. Assume $\mathcal{L}_1 = (L_1, |.|_1)$, $\mathcal{L}_2 = (L_2, |.|_2)$. Let

$$L := \{t\langle s_1, \dots, s_{\mathrm{rk}_a(t)}\rangle \mid t \in L_1,\ s_1, \dots, s_{\mathrm{rk}_a(t)} \in L_2\}$$

(as formal language over the alphabet $L_1 \cup L_2 \cup \{\langle, \rangle\} \cup \{,\}$) and define

$$|t\langle s_1, \dots, s_{\mathrm{rk}_a(t)}\rangle| := |t|_1 \circ_a \langle |s_1|_2, \dots, |s_{\mathrm{rk}_a(t)}|_2\rangle.$$

Then, by 2.16, $(L, |.|) \cong \mathcal{L}_1 \cdot_a \mathcal{L}_2$.

Let $t \in \mathrm{T}_\Sigma$, $t_1, \dots, t_{\mathrm{rk}_a(t)} \in \mathrm{T}_\Sigma$. Define

$$L_{t;t_1,\dots,t_{\mathrm{rk}_a(t)}} := \{s\langle s_1, \dots, s_{\mathrm{rk}_a(s)}\rangle \mid s \in L_1,\ s_i \in \mathcal{L}_2,\ \mathrm{ut}(|s|_1) = t,$$
$$\mathrm{ut}(|s_i|_2) = t_i,\ i = 1, \dots, \mathrm{rk}_a(s)\}.$$

On the other hand we define for $t \in \mathrm{T}_\Sigma$:

$$L_t := \{s \in L \mid \mathrm{ut}(|s|) = t\}.$$

Then obviously

$$L_t = \bigcup_{\substack{r \in \mathrm{T}_\Sigma \\ r_1,\dots,r_{\mathrm{rk}_a(r)} \in \mathrm{T}_\Sigma \\ r \circ_a \langle r_1,\dots,r_{\mathrm{rk}_a(r)}\rangle = t}} L_{r;r_1,\dots,r_{\mathrm{rk}_a(r)}}$$

and since the union in this definition is disjoint, we obtain

$$\begin{aligned}
(S_{\mathcal{L}_1 \cdot_a \mathcal{L}_2}, t) &= \bigoplus_{s \in L_t} \mathrm{wt}(|s|) = \bigoplus_{\substack{r \in \mathrm{T}_\Sigma \\ r_1,\dots,r_{\mathrm{rk}_a(r)} \in \mathrm{T}_\Sigma \\ r \circ_a \langle r_1,\dots,r_{\mathrm{rk}_a(r)}\rangle = t}} \bigoplus_{s \in L_{r;r_1,\dots,r_{\mathrm{rk}_a(r)}}} \mathrm{wt}(|s|) \\
&= \bigoplus_{\substack{r \in \mathrm{T}_\Sigma \\ r_1,\dots,r_{\mathrm{rk}_a(r)} \in \mathrm{T}_\Sigma \\ r \circ_a \langle r_1,\dots,r_{\mathrm{rk}_a(r)}\rangle = t}} \bigoplus_{\substack{s \in L_1 \\ \mathrm{ut}(|s|_1) = r}} \bigoplus_{\substack{s_1,\dots,s_{\mathrm{rk}_a(r)} \in L_2 \\ \mathrm{ut}(|s_i|_2) = r_i \\ i=1,\dots,\mathrm{rk}_a(r)}} \left(\mathrm{wt}(|s|_1) \odot \bigodot_{j=1}^{\mathrm{rk}_a(r)} \mathrm{wt}(|s_j|_2) \right) \quad \text{by 7.4} \\
&= \bigoplus_{\substack{r \in \mathrm{T}_\Sigma \\ r_1,\dots,r_{\mathrm{rk}_a(r)} \in \mathrm{T}_\Sigma \\ r \circ_a \langle r_1,\dots,r_{\mathrm{rk}_a(r)}\rangle = t}} \bigoplus_{\substack{s \in L_1 \\ \mathrm{ut}(|s|_1) = r}} \left[\mathrm{wt}(|s|_1) \odot \left(\bigoplus_{\substack{s_1,\dots,s_{\mathrm{rk}_a(r)} \in L_2 \\ \mathrm{ut}(|s_i|_2) = r_i \\ i=1,\dots,\mathrm{rk}_a(r)}} \bigodot_{j=1}^{\mathrm{rk}_a(r)} \mathrm{wt}(|s_j|_2) \right) \right] \quad \text{(distrib.)} \\
&= \bigoplus_{\substack{r \in \mathrm{T}_\Sigma \\ r_1,\dots,r_{\mathrm{rk}_a(r)} \in \mathrm{T}_\Sigma \\ r \circ_a \langle r_1,\dots,r_{\mathrm{rk}_a(r)}\rangle = t}} \left(\bigoplus_{\substack{s \in L_1 \\ \mathrm{ut}(|s|_1) = r}} \mathrm{wt}(|s|_1) \right) \odot \left(\bigodot_{j=1}^{\mathrm{rk}_a(r)} \bigoplus_{\substack{s_j \in L_2 \\ \mathrm{ut}(|s_j|_2) = r_j}} \mathrm{wt}(|s_j|_2) \right) \quad \text{(distrib.)} \\
&= \bigoplus_{\substack{r \in \mathrm{T}_\Sigma \\ r_1,\dots,r_{\mathrm{rk}_a(r)} \in \mathrm{T}_\Sigma \\ r \circ_a \langle r_1,\dots,r_{\mathrm{rk}_a(r)}\rangle = t}} (S_1, r) \odot \bigodot_{j=1}^{\mathrm{rk}_a(r)} (S_2, r_j) \\
&= (S_{\mathcal{L}_1} \cdot_a S_{\mathcal{L}_2}, t).
\end{aligned}$$

$\square$

7.15 Topcatenation. Let $f \in \Sigma^{(n)}$, $c \in K$, $S_1, \ldots, S_n \in \mathsf{FTS}_\Sigma$. Then we define $[f|c]\langle S_1, \ldots, S_n \rangle$ according to

$$([f|c]\langle S_1, \ldots, S_n\rangle, t) = \begin{cases} c \odot \bigodot_{i=1}^{n} (S_i, t_i) & \text{if } t = f\langle t_1, \ldots, t_n\rangle \\ 0 & \text{otherwise.} \end{cases}$$

7.16 Lemma. *For $f \in \Sigma^{(0)}$, $c \in K$ and $\mathcal{L}_1, \ldots, \mathcal{L}_n \in \mathit{WTL}_\Sigma$ we have*

$$S_{[f|c]\langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle} = [f|c]\langle S_{\mathcal{L}_1}, \ldots, S_{\mathcal{L}_n}\rangle.$$

Proof. Looking back at the definition of topcatenation in 2.8 we see that

$$(S_{[f|c]\langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle}, t) = \bigoplus_{\substack{(s_1, \ldots, s_n) \in L_1 \times \cdots \times L_n \\ \mathrm{ut}(|s_1, \ldots, s_n|) = t}} \mathrm{wt}(|s_1, \ldots, s_n|) \tag{*}$$

Supposing that $t = f\langle t_1, \ldots, t_n\rangle$ we obtain

$$\begin{aligned} &\bigoplus_{\substack{(s_1, \ldots, s_n) \in L_1 \times \cdots \times L_n \\ \mathrm{ut}(|s_1, \ldots, s_n|) = t}} \mathrm{wt}(|s_1, \ldots, s_n|) \\ &\quad = \bigoplus_{\substack{(s_1, \ldots, s_n) \in L_1 \times \cdots \times L_n \\ \mathrm{ut}(|s_i|_i) = t_i \\ i = 1, \ldots, n}} c \odot \bigodot_{j=1}^{n} \mathrm{wt}(|s_j|_j) \\ &\quad = c \odot \bigoplus_{\substack{s_1 \in L_1 \\ \mathrm{ut}(|s_1|_1) = t_1}} \cdots \bigoplus_{\substack{s_n \in L_n \\ \mathrm{ut}(|s_n|_n) = t_n}} \bigodot_{j=1}^{n} \mathrm{wt}(|s_j|_j) \qquad \text{(distrib.)} \\ &\quad = c \odot \bigoplus_{\substack{s_1 \in L_1 \\ \mathrm{ut}(|s_1|_1) = t_1}} \mathrm{wt}(|s_1|_1) \odot \bigoplus_{\substack{s_2 \in L_2 \\ \mathrm{ut}(|s_2|_2) = t_2}} \mathrm{wt}(|s_2|_2) \odot \ldots \odot \bigoplus_{\substack{s_n \in L_n \\ \mathrm{ut}(|s_n|_n) = t_n}} \mathrm{wt}(|s_n|_n) \quad \text{(distrib.)} \\ &\quad = c \odot \bigodot_{j=1}^{n} (S_j, s_j) \qquad \text{(distrib.)} \\ &\quad = ([f|c]\langle S_1, \ldots, S_n\rangle, t). \end{aligned}$$

If there are no $t_1, \ldots, t_n \in \mathrm{T}_\Sigma$ such that $t = f\langle t_1, \ldots, t_n\rangle$ then there can also be no $(s_1, \ldots, s_n) \in L_1 \times \cdots \times L_n$ with $\mathrm{ut}(|s_1, \ldots, s_n|) = t$ since

$$\mathrm{ut}(|s_1, \ldots, s_n|) = f\langle \mathrm{ut}(|s_1|_1), \ldots, \mathrm{ut}(|s_n|_n)\rangle.$$

Hence, in this case the sum in (*) is over the empty set and therefore it is equal to 0. $\square$

7.17 a-Iteration. Let $S \in \mathsf{FTS}_\Sigma$ and let $a \in \Sigma^{(0)}$ such that $(S, a) = 0$. Let $(\mathrm{T}_\Sigma^*, \mathrm{rk}, \circ, \varepsilon)$ be the free ranked monoid generated by $(\mathrm{T}_\Sigma, \mathrm{rk}_a)$ (cf. 1.20). Recall that $(\mathrm{T}_\Sigma, \mathrm{rk}_a, \circ_a, a)$ is a ranked monoid (cf. 1.21) and let

$$\varphi : (\mathrm{T}_\Sigma^*, \mathrm{rk}, \circ, \varepsilon) \longrightarrow (\mathrm{T}_\Sigma, \mathrm{rk}_a, \circ_a, a)$$

be the unique homomorphism induced by the identity map of T_Σ. On T_Σ^* we define a weight-function wt_S^* inductively by:

$$\mathrm{wt}_S^*(s) := \begin{cases} 1 & s = \varepsilon \\ (S, s) & s \in \mathrm{T}_\Sigma \\ (S, t) \odot \bigodot\limits_{i=1}^{\mathrm{rk}_a(t)} \mathrm{wt}_S^*(t_i) & s = t\langle t_1, \ldots, t_{\mathrm{rk}_a(t)}\rangle,\ t \in \mathrm{T}_\Sigma,\ t_1, \ldots, t_{\mathrm{rk}_a(t)} \in \mathrm{T}_\Sigma^*. \end{cases}$$

Then we define $S_a^* \in \mathsf{FTS}_\Sigma$ by

$$(S_a^*, t) := \bigoplus_{\substack{s \in \mathrm{T}_\Sigma^* \\ \varphi(s) = t}} \mathrm{wt}_S^*(s).$$

7.18 Lemma. *Assume, K is commutative. Let $a \in \Sigma^{(0)}$. Then for any finitary, a-quasiregular $\mathcal{L} \in \mathsf{WTL}_\Sigma$, we have*

$$S_{\mathcal{L}_a^*} = (S_\mathcal{L})_a^*.$$

Proof. We are going to use 2.23 that describes the a-iteration using ranked monoids. Let $(L_a^*, \mathrm{rk}, \circ, \varepsilon)$ be the free ranked monoid freely generated by (L, rk_a). The mapping $\psi : L \longrightarrow \mathrm{T}_\Sigma^*$, where $s \mapsto \mathrm{ut}(|s|)$, induces a unique homomorphism

$$\psi^\# : (L_a^*, \mathrm{rk}, \circ, \varepsilon) \longrightarrow (\mathrm{T}_\Sigma^*, \mathrm{rk}, \circ, \varepsilon).$$

Let $\chi : L \longrightarrow \mathrm{T}_\Sigma$ such that $s \mapsto \mathrm{ut}(|s|)$ and let $\chi^\#$ be the unique induced homomorphism from $(L_a^*, \mathrm{rk}, \circ, \varepsilon)$ to $(\mathrm{T}_\Sigma, \mathrm{rk}_a, \circ_a, a)$. Then the following diagram commutes since $U(\varphi^\#) \circ \psi = \chi$ where U is the forgetful functor mapping each ranked monoid to its carrier:

$$\begin{tikzcd} (\mathrm{T}_\Sigma^*, \mathrm{rk}, \circ, \varepsilon) \arrow[r, "\varphi^\#"] & (\mathrm{T}_\Sigma, \mathrm{rk}_a, \circ_a, a) \\ (L_a^*, \mathrm{rk}, \circ, \varepsilon) \arrow[u, "\psi^\#"] \arrow[ur, "\chi^\#"'] & \end{tikzcd}$$

In the following we show that

$$\mathrm{wt}_{S_\mathcal{L}}^*(t) = \bigoplus_{\substack{s \in L_a^* \\ \psi^\#(s) = t}} \mathrm{wt}(|s|_a^*)$$

for each $t \in \mathrm{T}_\Sigma^*$. This is done by induction on the structure of t.

If $t = \varepsilon$, then the only preimage of t under $\psi^\#$ is ε, hence $\mathrm{wt}^*(t) = 1 = \mathrm{wt}(|\varepsilon|_a^*)$. If $t \in \mathrm{T}_\Sigma$ then all preimages of t under $\psi^\#$ lie in L. Hence

$$\mathrm{wt}^*_{S_\mathcal{L}}(t) = (S_\mathcal{L}, t) = \bigoplus_{\substack{s \in \mathcal{L} \\ \mathrm{ut}(|s|)=t}} \mathrm{wt}(|s|)$$

and since $|.| = (|.|_a^*)_{|L}$ and $\mathrm{ut}(|.|)_{|L} = (\psi^\#)_{|L}$, we conclude

$$\mathrm{wt}^*_{S_\mathcal{L}}(t) = \bigoplus_{\substack{s \in L_a^* \\ \psi^\#(s)=t}} \mathrm{wt}(|s|_a^*).$$

If $t = r\langle r_1, \dots, r_{\mathrm{rk}_a(r)}\rangle$, where $r \in \mathrm{T}_\Sigma$ and $r_1, \dots, r_{\mathrm{rk}_a(r)} \in \mathrm{T}_\Sigma^*$, then

$$\begin{aligned}
\bigoplus_{\substack{s \in L_a^* \\ \psi^\#(s)=t}} \mathrm{wt}(|s|_a^*) &= \bigoplus_{\substack{s \in L \\ \mathrm{ut}(|s|)=r}} \bigoplus_{\substack{s_1,\dots,s_{\mathrm{rk}_a(r)} \in L_a^* \\ \psi^\#(s_i)=r_i \\ i=1,\dots,\mathrm{rk}_a(r)}} \mathrm{wt}(|s\langle s_1, \dots, s_{\mathrm{rk}_a(r)}\rangle|_a^*) \\
&= \bigoplus_{\substack{s \in L \\ \mathrm{ut}(|s|)=r}} \bigoplus_{\substack{s_1,\dots,s_{\mathrm{rk}_a(r)} \in L_a^* \\ \psi^\#(s_i)=r_i \\ i=1,\dots,\mathrm{rk}_a(r)}} \mathrm{wt}(|s| \circ_a \langle |s_1|_a^*, \dots, |s_{\mathrm{rk}_a(r)}|_a^*\rangle) \\
&= \bigoplus_{\substack{s \in L \\ \mathrm{ut}(|s|)=r}} \bigoplus_{\substack{s_1,\dots,s_{\mathrm{rk}_a(r)} \in L_a^* \\ \psi^\#(s_i)=r_i \\ i=1,\dots,\mathrm{rk}_a(r)}} \mathrm{wt}(|s|) \odot \bigodot_{j=1}^{\mathrm{rk}_a(r)} \mathrm{wt}(|s_j|_a^*) \qquad \text{by 7.4} \\
&= \bigoplus_{\substack{s \in L \\ \mathrm{ut}(|s|)=r}} \mathrm{wt}(|s|) \odot \left(\bigoplus_{\substack{s_1,\dots,s_{\mathrm{rk}_a(r)} \in L_a^* \\ \psi^\#(s_i)=r_i \\ i=1,\dots,\mathrm{rk}_a(r)}} \bigodot_{j=1}^{\mathrm{rk}_a(r)} \mathrm{wt}(|s_j|_a^*) \right) \qquad \text{(distrib.)} \\
&= (S, r) \odot \bigodot_{j=1}^{\mathrm{rk}_a(r)} \bigoplus_{\substack{s_j \in L_a^* \\ \psi^\#(s_j)=r_j}} \mathrm{wt}(|s_j|_a^*) \qquad \text{(distrib.)} \\
&= (S, r) \odot \bigodot_{j=1}^{\mathrm{rk}_a(r)} \mathrm{wt}^*_{S_\mathcal{L}}(r_j) \\
&= \mathrm{wt}^*_{S_\mathcal{L}}(t).
\end{aligned}$$

Finally we compute:

$$
\begin{aligned}
(S_{\mathcal{L}_a^*}, t) &= \bigoplus_{\substack{s \in L_a^* \\ \mathrm{ut}(|s|_a^*) = t}} \mathrm{wt}(|s|_a^*) \\
&= \bigoplus_{\substack{s \in L_a^* \\ \chi^{\#}(s) = t}} \mathrm{wt}(|s|_a^*) \\
&= \bigoplus_{\substack{r \in \mathrm{T}_\Sigma^* \\ \varphi^{\#}(r) = t}} \bigoplus_{\substack{s \in L_a^* \\ \psi^{\#}(s) = r}} \mathrm{wt}(|s|_a^*) \\
&= \bigoplus_{\substack{r \in \mathrm{T}_\Sigma^* \\ \varphi^{\#}(r) = t}} \mathrm{wt}^*_{S_{\mathcal{L}}}(r) \\
&= ((S_{\mathcal{L}})_a^*, t).
\end{aligned}
$$

□

7.19 a-Annihilation. Let $S \in \mathsf{FTS}_\Sigma$ and let $a \in \Sigma^{(0)}$. Then we define $S_{\neg a} \in \mathsf{FTS}_\Sigma$ according to

$$
(S_{\neg a}, t) := \begin{cases} (S, t) & \mathrm{rk}_a(t) = 0 \\ 0 & \text{else.} \end{cases}
$$

7.20 Lemma. *For any finitary $\mathcal{L} \in \mathsf{WTL}_\Sigma$ we have*

$$
S_{\mathcal{L}_{\neg a}} = (S_{\mathcal{L}})_{\neg a}.
$$

Proof. By 2.25 we have $\mathcal{L} \cong (L_{\neg a}, |.|_{\neg a})$ where $L_{\neg a} = \{s \in L \mid \mathrm{rk}_a(s) = 0\}$. Now

$$
(S_{\mathcal{L}_{\neg a}}, t) = \bigoplus_{\substack{s \in L_{\neg a} \\ \mathrm{ut}(|s|_{\neg a}) = t}} \mathrm{wt}(|s|_{\neg a}).
$$

If $\mathrm{rk}_a(t) \neq 0$ then this sum ranges over the empty set and hence it is equal to 0. If $\mathrm{rk}_a(t) = 0$, then

$$
\bigoplus_{\substack{s \in L_{\neg a} \\ \mathrm{ut}(|s|_{\neg a}) = t}} \mathrm{wt}(|s|_{\neg a}) = \bigoplus_{\substack{s \in L \\ \mathrm{ut}(|s|) = t}} \mathrm{wt}(|s|) = (S, t).
$$

□

7.21 a-Recursion. Let $S \in \mathsf{FTS}_\Sigma$ and let $a \in \Sigma^{(0)}$. Then we define $S_a^\mu := (S_a^*)_{\neg a}$.

7.22 Lemma. *Let $a \in \Sigma^{(0)}$. Then for any finitary, a-quasiregular $\mathcal{L} \in \mathsf{WTL}_\Sigma$ we have*

$$S_{\mathcal{L}_a^\mu} = (S_\mathcal{L})_a^\mu.$$

Proof. Clear, since $(S_\mathcal{L})_a^\mu = [(S_\mathcal{L})_a^*]_{\neg a} = (S_{\mathcal{L}_a^*})_{\neg a} = S_{(\mathcal{L}_a^*)_{\neg a}} = S_{\mathcal{L}_a^\mu}$. □

7.23 FTS-semantics of expressions. The previous definitions of operations on formal tree-series and their close relation to the corresponding operations on weighted tree-languages allows us to give FTS-semantics to most of the expressions that were introduced in Sections 5 and 6.

$$\begin{aligned}
(\!|a|\!) &:= 1 \cdot a && (a \in \Sigma^{(0)}),\\
(\!|x|\!) &:= 1 \cdot x && (x \in X),\\
(\!|f\langle e_1, \dots, e_n\rangle|\!) &:= [f|1]\langle (\!|e_1|\!), \dots, (\!|e_n|\!)\rangle && (f \in \Sigma,\ \mathrm{rk}(f) = n),\\
(\!|c \cdot e|\!) &:= c \cdot (\!|e|\!) && (c \in K),\\
(\!|e_1 + e_2|\!) &:= (\!|e_1|\!) + (\!|e_2|\!),\\
(\!|e_1 \cdot_x e_2|\!) &:= (\!|e_1|\!) \cdot_x (\!|e_2|\!) && (x \in X),\\
(\!|(e)_x^*|\!) &:= \begin{cases} (\!|e|\!)_x^* & \text{if } (\!|e|\!) \text{ is } x\text{-quasiregular}\\ \text{undef.} & \text{else}\end{cases} && (x \in X),\\
(\!|\mu x.(e)|\!) &:= \begin{cases} (\!|e|\!)_x^\mu & \text{if } (\!|e|\!) \text{ is } x\text{-quasiregular}\\ \text{undef.} & \text{else}\end{cases} && (x \in X).
\end{aligned}$$

7.24 Remark. Our FTS-semantics of expressions only produces formal tree-series from $\mathsf{FTS}_{\Sigma(X)}$. If we would like to talk about formal tree-series from $\mathsf{FTS}_{\Sigma(X_n)}$ then we need to do an identification of them with certain series from $\mathsf{FTS}_{\Sigma(X)}$. We chose this identification to be compatible with the corresponding embedding from $\mathsf{WTL}_{\Sigma(X_n)}$ into $\mathsf{WTL}_{\Sigma(X)}$. Given $S \in \mathsf{FTS}_{\Sigma(X_n)}$ we define $E(S) \in \mathsf{FTS}_{\Sigma(X)}$ by

$$(E(S), t) := \begin{cases} (S, t) & t \in \mathrm{T}_{\Sigma(X_n)}\\ 0 & \text{else.}\end{cases}$$

We call S *definable* in a set of rules $\mathcal{R}$ if $E(S)$ is definable in $\mathcal{R}$.

Now we are ready to translate 6.9 to formal tree-series:

7.25 Theorem. *Let K be commutative. Let $X = (x_i)_{i\in\mathbb{N}}$ be a family of distinct variable symbols disjoint from Σ. Let $S \in \mathsf{FTS}_{\Sigma(X)}$ and let*

$$\mathcal{R} := \{\mathrm{Cons}_a, \mathrm{Var}_x, \mathrm{Top}_f, \mathrm{Scal}_c, \mathrm{Sum} \mid c \in K, x \in X, a \in \Sigma^{(0)}\}.$$

Then the following are equivalent:

1. *S is recognizable,*
2. *S is definable through*

$$\mathcal{R} \cup \{\mathrm{pMu}_x \mid x \in X\},$$

3. *S is definable through*

$$\mathcal{R} \cup \{\mathrm{Prod}_x, \mathrm{pStar}_x \mid x \in X\},$$

4. *S is definable through*

$$\mathcal{R} \cup \{\mathrm{Neg}_x, \mathrm{pStar}_x \mid x \in X\}.$$

Proof. We only need to show that the rule Zero can be simulated by the others in frames of FTS-semantics. The rest is an immediate consequence of 6.9, 7.10, 7.12, 7.14, 7.16, 7.18, 7.20 and 7.22. But the Zero rule may always be simulated e.g. by

$$\mathrm{Var}_x \frac{}{\mathrm{Scal}_0 \dfrac{x}{0 \cdot x}}$$

□

7.26 Corollary. *Let K be commutative. Let $X = (x_i)_{i \in \mathbb{N}}$ be a family of distinct variable symbols disjoint from Σ. Then the set of all recognizable formal tree-series over $\Sigma(X)$ is the smallest subset of $\mathsf{FTS}_{\Sigma(X)}$ that contains all polynomials and that is closed with respect to sum, x-product ($x \in X$) and either x-iteration or x-recursion ($x \in X$) where the x-iteration and x-recursion operations are restricted to x-quasiregular series.*

Proof. This is a direct consequence of the previous theorem and 6.10. □

7.27 Remark. This corollary generalizes the Theorems 4.1 and 5.10 from [18]. Moreover, The equivalence in 7.25 between 1 and 2, generalizes Corollary 4.6 of [22] and the Kleene-type result from [36].

8 Fixed Point Theory (Preliminaries)

It is impossible to work on Kleene-type theorems and to ignore the field of iteration theories since these theories really seem to grasp the essential idea of such theorems. A substantial body of results from this area is collected in [4]. This is the first of three sections in this thesis, that deal with fixed point theory in the context of weighted tree-languages and formal tree-series.

Our first goal is to make a precise connection of the classical automata theoretic techniques from the previous sections with the fixed point theoretic techniques that were used by Kuich [36] and Bloom, Ésik [6] in their Kleene-type results for formal tree-series. Such a connection is essential if we want to claim that our results properly generalize their Kleene-type theorems.

Another reason to develop the fixed point theoretical aspects of weighted tree-languages is our wish to generalize a theorem by Berstel and Reutenauer which states that the recognizable formal tree-series are precisely those that appear as components of unique solutions of proper linear systems of equations. Fixed-point theory is the most elegant tool to reach this goal.

Last but not least fixed point theory for weighted tree-languages opens the door for further research on solving systems of equations, such as algebraic systems. It seems that some of the existing results in this area put too many restriction on the coefficient semiring.

In this section we introduce the essential notions from fixed point theory as far as we need them for the subsequent sections. Our exposition of iteration theories closely follows the book [4] by Bloom and Ésik.

8.1 Lawvere algebraic theories. A *Lawvere algebraic theory* (cf. [40]) is a pair $(T, (i_n)_{n\in\mathbb{N}})$ where T is a category whose objects are the non-negative integers and where (i_n) ($n \in \mathbb{N}$, $1 \le i \le n$) is a family of *distinguished morphisms* of T such that $i_n : 1 \longrightarrow n$, 1_1 is the identity morphism of 1 and for all objects n and m and for each family $f_i : 1 \longrightarrow m$ ($i = 1, \ldots, n$) there exist a unique arrow $f : n \longrightarrow m$ such that $f \circ i_n = f_i$ for all $i = 1, \ldots, n$. This arrow will be denoted by $\langle f_1, \ldots, f_n\rangle$ and is called the *(source-) tupling* of the f_i ($i = 1, \ldots, n$).

8.2 Some basic notions. The definition above implies that each object n of T is the n-th copower of 1 and that the morphisms $i_n : 1 \longrightarrow n$ are the corresponding copower-injections. An empty tupling of morphisms $1 \longrightarrow m$ yields a necessarily unique morphism $0_m : 0 \longrightarrow m$. In other words 0 is an initial object of T. The property that 1_1 is the unit of 1 ensures among other things that $\langle f\rangle = f$. Clearly, the unit $\mathbf{1}_n$ of n is equal to $\langle 1_n, \ldots, 1_n\rangle$.

Assume we are given a morphism $f : n \longrightarrow m$. Then we may define $f_i := f \circ i_n$. Evidently, then $f = \langle f_1, \ldots, f_n\rangle$.

The morphisms of T with domain 1 are called *scalar morphisms*. An arrow is called *base morphism* if it is either distinguished or it is a tupling of distinguished morphisms. For $n, p \in \mathbb{N}$, and $\varphi : \{1, \ldots, n\} \longrightarrow \{1, \ldots, m\}$ we may define a base-morphism $[\varphi] := \langle \varphi(1)_m, \ldots \varphi(n)_m \rangle : n \longrightarrow m$. Obviously every base-morphism may be obtained in this way. A base morphism is called *surjective* (*injective*) if it can be obtained as $[\varphi]$ for surjective (injective) φ. A theory is called nontrivial if the assignment $\varphi \mapsto [\varphi]$ is injective. In the sequel we will only work with nontrivial theories.

For mnemonic reasons (and to be consistent with the notions from [4]) we define for morphisms $f : n \longrightarrow m$, $g : m \longrightarrow p$: $f \cdot g := g \circ f : n \longrightarrow p$.

For $f : 1 \longrightarrow n$ and $g_1, \ldots, g_n : 1 \longrightarrow m$ we call the composition $f \cdot \langle g_1, \ldots, g_n \rangle$ *scalar composition* of f with $g_1, \ldots, g_m$.

8.3 Remark. The scalar morphisms of a Lawvere algebraic theory with codomain $\neq 0$ form an abstract clone. The projections of this clone are just the i_n and given $f : 1 \longrightarrow n$ and $g_1, \ldots, g_n : 1 \longrightarrow m$ the composition of f with $g_1, \ldots, g_n$ is defined by $f \cdot \langle g_1, \ldots, g_n \rangle : 1 \longrightarrow m$. The usual axioms

- $f \cdot \langle 1_n, \ldots, n_n \rangle = f$,
- $i_n \cdot \langle f_1, \ldots, f_n \rangle = f_i$,
- $f \cdot \langle g_1, \ldots, g_n \rangle \cdot \langle h_1, \ldots, h_k \rangle = f \cdot \langle g_1 \cdot \langle h_1, \ldots, h_k \rangle, \ldots, g_n \cdot \langle h_1, \ldots, h_k \rangle \rangle$

are easily verified. Indeed the last one follows from

$$\begin{aligned} i_m \cdot (\langle g_1, \ldots, g_n \rangle \cdot \langle h_1, \ldots, h_k \rangle) &= (i_m \cdot \langle g_1, \ldots, g_n \rangle) \cdot \langle h_1, \ldots, h_k \rangle \\ &= g_i \cdot \langle h_1, \ldots, h_k \rangle. \end{aligned}$$

Since on the other hand each arrow of the theory is uniquely expressible as the tupling of scalar arrows, the above mentioned abstract clone determines most of the algebraic theory (except the morphisms with codomain 0). For further details about abstract clones see e.g. [48].

8.4 Source pairing and separated sum. Let T be a theory. From the fact that every object of T is a copower of 1 it follows that T has coproducts. Indeed the coproduct of objects n and m is just $n + m$.

The injections $\kappa : n \longrightarrow n + m$ and $\lambda : m \longrightarrow n + m$ may be given by $\langle 1_{n+m}, \ldots, n_{n+m} \rangle$ and $\langle (n+1)_{n+m}, \ldots, (n+m)_{n+m} \rangle$, respectively. The universal property of the coproduct has the consequence that for any $p \geq 0$ and for any $f : n \longrightarrow p$, $g : m \longrightarrow p$ there exists a unique arrow $\langle f, g \rangle : n + m \longrightarrow p$ such that $f = \kappa \cdot \langle f, g \rangle$ and $g = \lambda \cdot \langle f, g \rangle$. The morphism $\langle f, g \rangle$ is called the *source-pairing* (or short: *pairing*) of f and g.

Given now $f : n \longrightarrow p$ and $g : m \longrightarrow q$, the *separated sum* $f \oplus g$ of f and g is defined by $\langle f \cdot \kappa, g \cdot \lambda \rangle$ where $\kappa : p \longrightarrow p + q$ and $\lambda : q \longrightarrow p + q$ are the coproduct-injections (cf. definition of source-pairing)

Note that the separated sum is in principle nothing else but the coproduct on morphisms. Moreover, do not confuse the sign $\oplus$ for theories with the operation of addition in semirings.

8.5 Example. Let $(L, \leq)$ be a complete partial order (cpo). That is, it is a partially ordered set in which each upwards directed set has a supremum. We define the theory $\mathrm{Th}(L, \leq)$ as follows: The morphisms from m to n are modeled by continuous functions from $(L, \leq)^n$ to $(L, \leq)^m$ (note the contravariance and recall that $(L, \leq)^n$, $(L, \leq)^m$ are again cpos and that a function is called continuous if it preserves the supremum of directed sets). The distinguished morphism $i_n : 1 \longrightarrow n$ is the usual projection to the i-th coordinate. The composition of arrows is the usual composition of functions. It is easy to see that $\mathrm{Th}(L, \leq)$ is indeed a theory.

8.6 T-morphisms, subtheories, congruences. Let T_1 and T_2 be two theories. A T-*morphism* $\varphi : T_1 \longrightarrow T_2$ is a functor that acts as identity on objects and that preserves the distinguished arrows. T_1 is called *subtheory* of T_2 (written $T_1 \leq T_2$) if T_1 is a subcategory of T_2 such that the corresponding inclusion functor is a T-morphism.

The concept of T-morphisms leads us immediately to the concept of congruences. Given a theory T a *theory congruence* $\approx$ of T is a family $(\approx_{n,m})_{n,m \in \mathbb{N}}$ of equivalence relations on each hom-set $T(n, m)$ $(n, m \in \mathbb{N})$ such that

1. $f \approx_{n,p} g,\ f' \approx_{p,m} g' \Rightarrow f \cdot f' \approx_{n,m} g \cdot g'$,
2. $f_i \approx_{1,m} g_i\,(i = 1, \ldots, n) \ \Rightarrow\ \langle f_1, \ldots, f_n \rangle \approx_{n,m} \langle g_1, \ldots, g_n \rangle$.

We will usually omit the subscripts from the $\approx$-sign.

For every arrow f of T let $[f]_\approx$ be the equivalence class of f. The quotient theory $T/{\approx}$ has as arrows all equivalence classes of morphisms from T where the distinguished morphism i_n of $T/{\approx}$ is $[i_n]_\approx$ (for $n \in \mathbb{N}$ and $1 \leq i \leq n$). Then $T/{\approx}$ is indeed a theory and the functor $\mathrm{nat}_\approx : T \longrightarrow T/{\approx}$ that maps every morphism f of T to $[f]_\approx$, is a T-morphism. On the other hand, if $\varphi : T \longrightarrow T'$ is a T-morphism, then we may define for $f, g \in T(n, m)$: $f \approx g :\Longleftrightarrow\ \varphi(f) = \varphi(g)$. This relation defines a congruence of T. It is also denoted by $\ker \varphi$. The usual homomorphism theorem holds: If $\varphi : T \longrightarrow T'$ is surjective on hom-sets, then $T' \cong T/\ker \varphi$.

Together with the above defined notion of T-morphisms the Lawvere algebraic theories form a category, denoted by TH.

8.7 Remark. We must be careful with the statement that TH is a category. In fact we never claimed that theories should be locally small. Indeed later on we shall work with theories in which the arrows from n to m form proper classes. But then it is not true that the theories form a class. This is the usual foundational dilemma of category theory. It may be overcome given a set-theory allowing for objects of higher order than classes such as e.g. conglomerates. However, this is way beyond the scope of this treatise where only a finite number of theories will ever be considered. Still we would like to stay in the notational realm of category theory. Therefore we will take the innocent point of view that "everything that has objects and morphisms that fulfill the usual axioms of category theory **is** a category". We just note that the category of theories is not large (in the usual categorial sense to have a proper class of objects) but very large.

8.8 Pre-iteration theories. A *pre-iteration theory* is a pair $(T, \dagger)$ where $T = (T, (i_n))$ is a theory and where $\dagger$ is a partial operation on the morphisms of T such that

$$\begin{aligned} \dagger \,:\, T(n, n+p) &\longrightarrow T(n,p) \qquad\qquad (n,p \in \mathbb{N}) \\ f &\mapsto f^\dagger. \end{aligned}$$

Morphisms of pre-iteration theories are T-morphisms that preserve the dagger-operation. If $\dagger$ is not important or if it is clear from the context, then instead of $(T, \dagger)$ we will usually write just T. The dagger-operation restricted to scalar arrows is called *scalar dagger*.

8.9 Example. Let $(L, \leq)$ be a complete partially ordered set (cpo). Let $\mathrm{Th}(L, \leq)$ be the theory of continuous functions on powers of $(L, \leq)$ (cf. 8.5). By the Knaster-Tarski theorem every continuous function $f : L^n \longrightarrow L^n$ has a least fixed point. Hence for each $f : L^{n+p} \longrightarrow L^n$ and for each $y \in L^p$ there is a least fixed point of $f_y : L^n \longrightarrow L^n$, $x \mapsto f(x,y)$. It shall be denoted by $f^\dagger(y)$. It is elementary to show that the function $f^\dagger : L^p \longrightarrow L^n$, $y \mapsto f^\dagger(y)$ is again continuous. Together with this dagger-operation $\mathrm{Th}(L, \leq)$ forms a pre-iteration theory.

8.10 Conway theories. A *Conway theory* is a pre-iteration theory that satisfies the following identities:

1. Left zero identity

$$(0_n \oplus f)^\dagger = f$$

   for all $f : n \longrightarrow p$,

2. Right zero identity

$$(f \oplus 0_q)^\dagger = f^\dagger \oplus 0_q$$

   for all $f : n \longrightarrow n+p$,

3. Pairing identity

$$\langle f, g\rangle^\dagger = \langle f^\dagger \cdot \langle h^\dagger, \mathbf{1}_p\rangle, h^\dagger\rangle$$

   for all $f : n \longrightarrow n+m+p$, $g : m \longrightarrow n+m+p$ where $h = g \cdot \langle f^\dagger, \mathbf{1}_{m+p}\rangle : m \longrightarrow m+p$,

4. Permutation identity

$$([\pi] \cdot f \cdot ([\pi^{-1}] \oplus \mathbf{1}_p)))^\dagger = [\pi] \cdot f^\dagger$$

for all $f : n \longrightarrow n + p$ and for all permutations π of the numbers $\{1, \ldots, n\}$ where $[\pi]$ is the corresponding base-morphisms (cf. 8.2).

8.11 Example. For any cpo $(L, \leq)$ the theory $\mathrm{Th}(L, \leq)$ is a Conway theory. A proof of this may be found e.g. in [4].

8.12 Remark. Several other interesting identities follow from the axioms of Conway theories. We just mention them here without proofs which can be found e.g. in [4]:

1. (Elgot) Fixed point identity

$$f^\dagger = f \cdot \langle f^\dagger, \mathbf{1}_p \rangle$$

for all $f : n \longrightarrow n + p$,

2. Parameter identity:

$$(f \cdot (\mathbf{1}_n \oplus g))^\dagger = f^\dagger \cdot g$$

for all $f : n \longrightarrow n + p$, $g : p \longrightarrow q$,

3. Composition identity

$$(f \cdot \langle g, 0_n \oplus \mathbf{1}_p \rangle)^\dagger = f \cdot \langle (g \cdot \langle f, 0_m, \mathbf{1}_p \rangle)^\dagger, \mathbf{1}_p \rangle$$

for all $f : n \longrightarrow m + p$, $g : m \longrightarrow n + p$,

4. Double dagger identity

$$f^{\dagger\dagger} = (f \cdot (\langle \mathbf{1}_n, \mathbf{1}_n \rangle + \mathbf{1}_p))^\dagger$$

for all $f : n \longrightarrow n + n + p$.

8.13 Lemma. *Given a Conway theory T and a theory $U \leq T$. If $f \in U$ implies $f^\dagger \in U$ for each scalar morphism f, then U is also a Conway theory.*

Proof. All we have to show is that for $f : n \longrightarrow n + p$ we have

$$f \in U \Rightarrow f^\dagger \in U. \qquad (*)$$

We do this by induction on n: For $f : 1 \longrightarrow 1 + p$ we obtain just the assumption of the Lemma. Suppose now that the implication (*) holds for all $f : n \longrightarrow n + p$.

Let $f : n+1 \longrightarrow n+1+p$. Then $f = \langle \hat{f}, g\rangle$ for $\hat{f} : n \longrightarrow n+1+p$ and $g : 1 \longrightarrow n+1+p$. By the pairing identity

$$f^\dagger = \langle \hat{f}, g\rangle^\dagger = \langle \hat{f}^\dagger \cdot \langle h^\dagger, \mathbf{1}_p\rangle, h^\dagger\rangle$$

where $h = g \cdot \langle \hat{f}^\dagger, \mathbf{1}_{1+p}\rangle : 1 \longrightarrow 1+p$. From $f \in U$ follows $\hat{f}, g \in U$. By the induction hypothesis $\hat{f}^\dagger \in U$. Hence $h \in U$. Again by the induction hypothesis $h^\dagger \in U$. Hence $\langle \hat{f}, g\rangle^\dagger = f^\dagger \in U$. □

8.14 Remark. Note that the proof above shows that the dagger of any arrow $f = \langle f_1, \dots, f_n\rangle : n \longrightarrow n+p$ can be obtained from its components and from the distinguished arrows by scalar composition and scalar dagger.

8.15 Iteration theories. Let T be a theory and let $f = \langle f_1, \dots, f_n\rangle : n \longrightarrow m+p$, $g_i : m \longrightarrow k$ be morphisms of T ($i = 1, \dots, n$). Then we define

$$f \parallel (g_1, \dots, g_n) := \langle f_1 \cdot (g_1 \oplus \mathbf{1}_p), \dots, f_n \cdot (g_n \oplus \mathbf{1}_p)\rangle : n \longrightarrow k+p.$$

A Conway theory $(T, \dagger)$ is called *iteration theory* if the commutative identity holds in T. That is

$$((\varrho \cdot f) \parallel (\varrho_1, \dots, \varrho_m))^\dagger = \varrho \cdot (f \cdot (\varrho \oplus \mathbf{1}_p))^\dagger$$

for all $f : n \longrightarrow m+p$, $\varrho : m \longrightarrow n$ surjective base, $\varrho_i : m \longrightarrow m$ base such that $\varrho_i \cdot \varrho = \varrho$ ($i = 1, \dots, m$)

8.16 Pre-iteration theories as heterogeneous algebras. Following [4, Chap. 3.3] pre-iteration theories can be considered as certain heterogeneous (many-sorted) algebras. The set S of sorts consists of all pairs (n, m) of natural numbers. As operation-symbols we take

$$\begin{aligned}
\cdot_{m,k,n} &: (m,k)(k,n) \longrightarrow (m,n) && (m,k,n \in \mathbb{N}),\\
\langle . \rangle_{m,n} &: \underbrace{(1,n)\cdots(1,n)}_{m \text{ times}} \longrightarrow (m,n) && (m,n \in \mathbb{N}),\\
(.)^\dagger_{n,p} &: (n, n+p) \longrightarrow (n,p) && (n,p \in \mathbb{N}).
\end{aligned}$$

Moreover we have constant-symbols i_n for $n \in \mathbb{N}$ and $1 \le i \le n$.

Given a pre-iteration theory. $(T, \dagger)$, the heterogeneous algebra associated with T has as carrier the family $(T_{n,m})_{(n,m)\in S}$ where $T_{n,m} := T(n,m)$—the collection of morphisms from n to m in T. Moreover

$$\begin{aligned}
\cdot_{m,k,n} &: T_{m,k} \times T_{k,m} \longrightarrow T_{m,n} : (f,g) \mapsto f \cdot g,\\
\langle . \rangle_{m,n} &: (T_{1,n})^m \longrightarrow T_{m,n} : (f_1, \dots, f_m) \mapsto \langle f_1, \dots, f_m\rangle,\\
(.)^\dagger_{n,p} &: T_{n,n+p} \longrightarrow T_{n,p} : f \mapsto f^\dagger.
\end{aligned}$$

Moreover the interpretation of the constants i_n are the corresponding distinguished morphisms of T. We will usually drop the indices from the operations if they are clear from the context.

8.17 Terms and valuations in pre-iteration theories. Terms in the signature given above can now be defined as follows: We are taking for each sort $(n,m) \in S$ a countably infinite set $F_{n,m}$ of variable symbols. Every term t to be defined will have its sort. If t has sort (n,m) then this will be denoted by $t : n \longrightarrow m$. The definition of the terms is done by structural induction: For $n \in \mathbb{N}$ and $1 \le i \le n$ we let $i_n : 1 \longrightarrow n$ be a term. If $f \in F_{n,m}$, then $f : n \longrightarrow m$ is a term. If $t_1 : m \longrightarrow k$ and $t_2 : k \longrightarrow n$ are terms, then $t_1 \cdot_{m,k,n} t_2 : m \longrightarrow n$ is a term. Moreover, if $t_1, \dots, t_m : 1 \longrightarrow n$ are terms, then $\langle t_1, \dots, t_m \rangle_{m,n} : m \longrightarrow n$ is a term. Finally, if $t : n \longrightarrow n+p$ is a term, then $(t)^\dagger_{n,p} : n \longrightarrow p$ is a term.

Given a pre-iteration theory T. Let $((T_{n,m}), (\cdot_{m,k,n}), (\langle . \rangle_{m,n}), ((.)^\dagger_{n,p}), (i_n))$ be its associated heterogeneous algebra. A *valuation* in T is a family of functions $V = (V_{n,m})_{(n,m)\in S}$ where $V_{n,m} : F_{n,m} \longrightarrow T_{n,m}$. Now to every term t we associate its value $[t]_V$ under the valuation V. As usual this is done by structural induction: For $n \in \mathbb{N}$ and $1 \le i \le n$ we define $[i_n]_V := i_n$. For $f \in F_{n,m}$ we define $[f]_V := V_{n,m}(f)$. For terms $t : m \longrightarrow k$ and $t' : k \longrightarrow n$ we define $[t \cdot_{m,k,n} t']_V := [t]_V \cdot [t']_V$. For $t_1, \dots, t_m : 1 \longrightarrow n$ we define $[\langle t_1, \dots, t_m \rangle_{m,n}]_V := \langle [t_1]_V, \dots, [t_m]_V \rangle$. Finally, for $t : n \longrightarrow n+p$ we define $[(t)^\dagger_{n,p}]_V := ([t]_V)^\dagger$.

Let us define several notational shortcuts for special terms: $\mathbf{1}_n := \langle 1_n, \dots, n_n \rangle_{n,n}$, $0_n := \langle \, \rangle_{0,n}$. For $t : n \longrightarrow m$ we define $t_i := i_n \cdot_{1,n,m} t$. For $t : n \longrightarrow p$ and $t' : m \longrightarrow p$ we define $\langle t, t' \rangle := \langle t_1, \dots, t_n, t'_1, \dots, t'_m \rangle_{n+m,p}$. Moreover, for $t : n \longrightarrow p$ and for $g : m \longrightarrow q$ we define $f \oplus g := \langle f \cdot_{n,p,p+q} \kappa, g \cdot_{m,q,p+q} \lambda \rangle$ where $\kappa = \langle 1_{p+q}, \dots, p_{p+q} \rangle_{p,p+q}$ and $\lambda = \langle (p+1)_{p+q}, \dots, (p+q)_{p+q} \rangle_{q,p+q}$. If the type of the operation is clear from the context, we write $\langle t_1, \dots, t_n \rangle$ instead of $\langle t_1, \dots, t_n \rangle_{n,m}$ for $t_i : 1 \longrightarrow m$ $(i = 1, \dots, n)$ and $t \cdot t'$ instead of $t \cdot_{m,k,n} t'$ for $t : m \longrightarrow k$ and $t' : k \longrightarrow n$ and finally $t^\dagger$ instead of $(t)^\dagger_{n,p}$ for $t : n \longrightarrow n+p$.

Let V be a valuation. Then for any $t : n \longrightarrow p$, $t' : m \longrightarrow p$ we have $[\langle t, t' \rangle]_V = \langle [t]_V, [t']_V \rangle$, and for all $t : n \longrightarrow p$, $t' : m \longrightarrow q$ we have $[t \oplus t']_V = [t]_V \oplus [t']_V$. Moreover $[0_n]_V = 0_n$ and $[\mathbf{1}_n]_V = \mathbf{1}_n$.

8.18 Identities in pre-iteration theories. An *identity* in a pre-iteration theory is a pair of two terms t and t' of the same sort. Identities are denoted by $t = t'$. Given a pre-iteration theory T, we say that the identity $t = t'$ *holds in* T (denoted by $T \vDash t = t'$) if for all valuations V in T we have $[t]_V = [t']_V$.

8.19 Lemma. *Let T and T' be pre-iteration theories such that there is an epimorphism from T to T'. Let $t = t'$ be an identity. Then $T \vDash t = t'$ implies $T' \vDash t = t'$.* □

8.20 Remark. According to [5], the iteration theories are precisely all those pre-iteration theories in which all identities valid in all cpo-theories hold.

8.21 Functor-theories. Let us now come to the introduction of the type of theories that will be of main interest to us (for many more examples of theories see [5]). Given a category C. We will identify the natural number n with the category

C^n. In particular the number 0 is associated to the trivial category that consists of one object and one morphism. As arrows from m to n we take functors from C^n to C^m. The distinguished arrow $i_n : 1 \longrightarrow n$ is modeled by the natural projection functor from $\mathsf{C}^n \longrightarrow \mathsf{C}$ that maps a tuple $(X_1, \dots, X_n)$ to X_i (and acts likewise on morphisms). It is easy to see that in this way we obtain a theory[6]. It is called the *functor-theory* of C and it will be denoted by $\mathrm{Th}(\mathsf{C})$.

8.22 Dagger-operation on functor-theories. Let now $F : \mathsf{C}^{n+p} \longrightarrow \mathsf{C}^n$ be a functor. For $Y \in \mathsf{C}^p$ an F_Y*-algebra* is a pair (X, f) where $X \in \mathsf{C}^n$ and $f : F(X,Y) \longrightarrow X$. Let (X', f') be another F_Y-algebra. A morphism $\varphi : X \longrightarrow X'$. is called F_Y*-homomorphism* from (X, f) to (X', f') if the following diagram commutes:

$$\begin{tikzcd}
F(X,Y) \arrow[r, "f"] \arrow[d, "F(\varphi, \mathbf{1}_Y)"'] & X \arrow[d, "\varphi"] \\
F(X',Y) \arrow[r, "f'"] & Y
\end{tikzcd}$$

Together with this notion of homomorphisms the F_y-algebras form a category. An initial object in this category is called *initial* F_Y*-algebra*. By the usual argument any two initial F_Y-algebras are isomorphic (if an initial F_Y-algebra exists at all) and the structure map of any initial algebra is an isomorphism.

Assume now that there exists an initial F_Y-algebra $(F^\dagger(Y), \mu_{F,Y})$ for each $Y \in \mathsf{C}^p$. Then the assignment $Y \mapsto F^\dagger(Y)$ may be extended uniquely to a functor $F^\dagger : \mathsf{C}^p \longrightarrow \mathsf{C}^n$ such that $\mu_F = (\mu_{F,Y})_{Y \in \mathsf{C}^p} : F \circ [F^\dagger, \mathbf{1}_p] \longrightarrow F^\dagger$ is a natural isomorphism where $[F^\dagger, \mathbf{1}_p]$ denotes the pairing of $F^\dagger$ and $\mathbf{1}_p$. The image of any morphism $f : Y \longrightarrow Y'$ from C^p under $F^\dagger$ is the initial F_Y-morphism from $(F^\dagger(Y), \mu_{F,Y})$ to $(F^\dagger(Y'), \mu_{F,Y'} \circ F(\mathbf{1}_n, f))$. The situation is summed up in the following diagram:

$$\begin{tikzcd}
F(F^\dagger(Y'),Y) \arrow[r, "F(\mathbf{1}_n,f)"] & F(F^\dagger(Y'),Y') \arrow[r, "\mu_{F,Y'}"] & F^\dagger(Y') \\
F(F^\dagger(Y),Y) \arrow[u, dashed, "F(!,\mathbf{1}_p)"] \arrow[rr, "\mu_{F,Y}"'] & & F^\dagger(Y) \arrow[u, dashed, "!=:F^\dagger(f)"']
\end{tikzcd}$$

Functoriality of this assignment is clear because of the uniqueness of the initial homomorphisms. Moreover the above drawn diagram exactly ensures that μ_F is a natural transformation. The fact that it is indeed a natural isomorphism follows easily from the fact that the $\mu_{F,Y}$ are iso.

8.23 Algebraically complete categories [4]. Given a category C with functor-theory $\mathrm{Th}(\mathsf{C})$. A pair $(\mathsf{C}, \mathcal{F})$ is called *algebraically complete category* if $\mathcal{F}$ is a subtheory of $\mathrm{Th}(\mathsf{C})$ such that for each $F : \mathsf{C}^{n+p} \longrightarrow \mathsf{C}^n$ from $\mathcal{F}$ and for each $Y \in \mathsf{C}^p$ there exists an initial F_Y-algebra $(F^\dagger(Y), \mu_{F,Y})$ such that the functor $F^\dagger$ is again in $\mathcal{F}$.

[6] Note that this theory will in general not be locally small

8.24 The theory of ω^{op}-continuous functors. It is known that for an ω-cocomplete category with initial object the ω^{op}-continuous functors on powers of C form a sub-theory of $\mathrm{Th}(\mathsf{C})$ that is algebraically complete (see e.g. [4]). This theory is denoted by $\mathrm{Th}_\omega(\mathsf{C})$.

The theories of ω^{op}-continuous functors were studied by Bloom and Ésik in [4]. They show that in such theories all identities of iteration theories hold "up to isomorphism". To be more precise, the natural isomorphism relation for functors is a theory congruence and the quotient-theory by this congruence is an iteration theory.

This result was later on greatly generalized by Ésik and Labella in [23]. They showed that the equational theory of algebraically complete categories is the same as the one for iteration theories.

9 Fixed Point Theory of Weighted Tree-Languages

In this section we examine the fixed point theoretical aspects of weighted tree-languages. In the iteration theory of all ω^{op}-continuous functors on powers of $\mathsf{WTL}_{\Sigma(X)}$ we identify a subtheory $\mathrm{WTh}_{\Sigma(X)}$ that can be seen as a theory of weighted tree-languages. That is, to each morphism of $\mathrm{WTh}_{\Sigma(X)}$ there corresponds a tuple of weighted tree-languages and vice-versa—up to equivalence. We show that this theory is in fact also an iteration theory. Moreover we show that the weakly recognizable weighted tree-languages define a sub iteration theory of $\mathrm{WTh}_{\Sigma(X)}$. This result is complemented by a characterization of recognizable and weakly recognizable weighted tree-languages by behaviors of quasiregular normal descriptions and normal descriptions, respectively. That in turn shows that our notion of weighted tree-automata matches the notion used by Bloom and Ésik in their work on formal tree-series. Finally we conclude an iteration-theoretical Kleene-type theorem for weakly recognizable weighted tree-languages. A similar characterization of the recognizable weighted tree-languages would be possible but since our main goal is to give such a description on the level of formal tree-series, we postpone the necessary arguments to the section on the fixed point theory of formal tree-series.

9.1 Lemma. *The Lawvere-theory* $\mathrm{Th}_\omega(\mathsf{WTL}_{\Sigma(X)})$ *of* ω^{op}*-continuous functors on powers of* $\mathsf{WTL}_{\Sigma(X)}$ *is algebraically complete and therefore an iteration theory "up to isomorphism".*

Proof. WTL_Σ has the empty weighted tree-language as an initial object and arbitrary colimits. Hence it is also ω-cocomplete. The rest follows from 8.24. □

It is very nice to have an iteration theory on top of $\mathsf{WTL}_{\Sigma(X)}$. However, what we would like to have is an iteration theory *of* weighted tree-languages. Still $\mathrm{Th}_\omega(\mathsf{WTL}_{\Sigma(X)})$ is a good start because we will identify the desired theory of weighted tree-languages as a subtheory of it. But before we are ready to define this subtheory, we need to introduce another concept of substitution on weighted trees—the OI-substitution:

9.2 OI-substitution into trees. Let $X = (x_i)_{i\in\mathbb{N}^+}$ be a family of distinct variable-symbols that is disjoint from Σ and define $X_i := \{x_1, \ldots, x_i\}$ $(i \in \mathbb{N})$. Let $t \in \mathrm{WT}_{\Sigma(X_n)}$, $\mathcal{L}_1, \ldots, \mathcal{L}_n \in \mathsf{WTL}_{\Sigma(X)}$. We define the *OI-substitution* $t \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle$ of $\mathcal{L}_1, \ldots, \mathcal{L}_n$ into t by induction on the structure of t:

- $[x_i|c] \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle := c \cdot \mathcal{L}_i \qquad (i = 1, \ldots, n),$
- $[a|c] \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle := \{[a|c]\} \qquad (a \in \Sigma^{(0)})$

- and if $f \in \Sigma^{(k)}$, $s_1, \ldots, s_k \in \mathrm{WT}_{\Sigma(X_n)}$, then

$$([f|c]\langle s_1, \ldots, s_k\rangle) \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle$$
$$:= [f|c]\langle s_1 \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle, \ldots, s_k \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle\rangle.$$

Thus every t determines a functor from $\mathsf{WTL}^n_{\Sigma(X)}$ to $\mathsf{WTL}_{\Sigma(X)}$. It is perfectly clear that the restriction of this functor to $\mathsf{WTL}_{\Sigma(X_m)}$ is welldefined for each $m \in \mathbb{N}$.

9.3 Lemma. *For $t \in \mathrm{WT}_{\Sigma(X_n)}$ the functor $(\mathcal{L}_1, \ldots, \mathcal{L}_n) \mapsto t \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle$ preserves directed colimits.*

Proof. The projection-functor, the constant functors and the topcatenation-functors preserve directed colimits. Moreover, functors that preserve directed colimits are closed under composition. □

9.4 OI-substitution. Like in 2.15 the OI-substitution into trees may be lifted to whole weighted tree-languages. Let $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X_n)}$, $\mathcal{L}_1, \ldots, \mathcal{L}_n \in \mathsf{WTL}_{\Sigma(X)}$. Then the functor

$$(\mathcal{L}_1, \ldots, \mathcal{L}_n) \mapsto \mathcal{L} \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle := \coprod_{t \in \mathcal{L}} |t| \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle$$

obviously preserves directed colimits.

According to 2.13 and 2.14, for given fixed $\mathcal{L}_1, \ldots, \mathcal{L}_n$ weighted tree-languages the assignment $\mathcal{L} \mapsto \mathcal{L} \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle$ is also functorial and preserves arbitrary colimits.

9.5 Lemma. *Let $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X_n)}$, $\mathcal{L}_1, \ldots, \mathcal{L}_n \in \mathsf{WTL}_{\Sigma(X_m)}$, and let $\mathcal{M}_1, \ldots, \mathcal{M}_m \in \mathsf{WTL}_{\Sigma(X)}$. Then*

1. $(c \cdot \mathcal{L}) \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle \cong c \cdot (\mathcal{L} \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle)$,
2. $[f|c]\langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle \cdot \langle \mathcal{M}_1, \ldots, \mathcal{M}_m\rangle$
$$\cong [f|c]\langle \mathcal{L}_1 \cdot \langle \mathcal{M}_1, \ldots, \mathcal{M}_m\rangle, \ldots, \mathcal{L}_n \cdot \langle \mathcal{M}_1, \ldots, \mathcal{M}_m\rangle\rangle,$$
3. $(\mathcal{L} \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle) \cdot \langle \mathcal{M}_1, \ldots, \mathcal{M}_m\rangle$
$$\cong \mathcal{L} \cdot \langle \mathcal{L}_1 \cdot \langle \mathcal{M}_1, \ldots, \mathcal{M}_m\rangle, \ldots, \mathcal{L}_n \cdot \langle \mathcal{M}_1, \ldots, \mathcal{M}_m\rangle\rangle.$$

Proof. **about 1.:** First we show that $(c \cdot |t|) \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle \cong c \cdot (|t| \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle)$ for $t \in \mathcal{L}$:

If $|t| = [x_i|d]$, then

$$\begin{aligned} c \cdot (|t| \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle) &= c \cdot (d \cdot \mathcal{L}_i) = [x_i|c \odot d] \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle \\ &= (c \cdot |t|) \cdot \langle \mathcal{L}_1, \ldots \mathcal{L}_n\rangle. \end{aligned}$$

If $|t| = [a|d]$ then

$$c \cdot (|t| \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle) = c \cdot \{[a|d]\} = \{[a|c \odot d]\} = (c \cdot t) \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n\rangle.$$

If $|t| = [f|d]\langle s_1, \dots, s_k\rangle$, then

$$\begin{aligned} c \cdot (|t| \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle) &= c \cdot [f|d]\langle s_1 \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle, \dots, s_k \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle\rangle \\ &= [f|c \odot d]\langle s_1 \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle, \dots, s_k \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle\rangle \\ &= (c \cdot |t|) \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle. \end{aligned}$$

Finally we argue that

$$\begin{aligned} c \cdot (\mathcal{L} \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle) &= c \cdot \coprod_{t \in \mathcal{L}} |t| \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle \\ &\cong \coprod_{t \in \mathcal{L}} c \cdot (|t| \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle) \\ &= \coprod_{t \in \mathcal{L}} (c \cdot |t|) \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle = (c \cdot \mathcal{L}) \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle. \end{aligned}$$

about 2:

$$\begin{aligned} &([f|c]\langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle) \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle \\ &\qquad = \coprod_{\substack{s_i \in \mathcal{L}_i \\ i=1,\dots,n}} [f|c]\langle |s_1|, \dots, |s_n|\rangle \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle \\ &\qquad = \coprod_{\substack{s_i \in \mathcal{L}_i \\ i=1,\dots,n}} [f|c]\langle |s_1| \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle, \dots, |s_n| \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle\rangle \end{aligned}$$

and since topcatenation preserves colimits:

$$\begin{aligned} &\cong [f|c]\langle \coprod_{s_1 \in \mathcal{L}_1} |s_1| \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle, \dots, \coprod_{s_n \in \mathcal{L}_n} |s_n| \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle\rangle \\ &= [f|c]\langle \mathcal{L}_1 \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle, \dots, \mathcal{L}_n \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle\rangle. \end{aligned}$$

about 3: Let $t \in \mathcal{L}$. First we show that

$$\begin{aligned} &(|t| \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle) \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle \\ &\qquad\qquad \cong |t| \cdot \langle \mathcal{L}_1 \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle, \dots, \mathcal{L}_n \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle\rangle. \end{aligned}$$

If $|t| = [x_i|c]$, then

$$\begin{aligned} &|t| \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle) \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle \\ &\qquad\qquad = (c \cdot \mathcal{L}_i) \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle \\ &\qquad\qquad \cong c \cdot (\mathcal{L}_i \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle) \qquad\qquad \text{(by 1.)} \\ &\qquad\qquad = |t| \cdot \langle \mathcal{L}_1 \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle, \dots, \mathcal{L}_n \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle\rangle. \end{aligned}$$

If $|t| = [a|c]$, then

$$\begin{aligned} &|t| \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n\rangle) \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle \\ &\qquad\qquad = \{[a|c]\} \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle = \{[a|c]\} \\ &\qquad\qquad = |t| \cdot \langle \mathcal{L}_1 \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle, \dots, \mathcal{L}_n \cdot \langle \mathcal{M}_1, \dots, \mathcal{M}_m\rangle\rangle. \end{aligned}$$

If $|t| = [f|c]\langle s_1, \dots , s_k\rangle$, then

$$\begin{aligned}
&(|t| \cdot \langle \mathcal{L}_1, \dots , \mathcal{L}_n\rangle) \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle \\
&\quad = [f|c]\langle s_1 \cdot \langle \mathcal{L}_1, \dots , \mathcal{L}_n\rangle, \dots , s_k \cdot \langle \mathcal{L}_1, \dots , \mathcal{L}_n\rangle\rangle \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle \\
&\quad \cong [f|c]\langle s_1 \cdot \langle \mathcal{L}_1, \dots , \mathcal{L}_n\rangle \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle, \dots , \\
&\qquad\qquad s_k \cdot \langle \mathcal{L}_1, \dots , \mathcal{L}_n\rangle \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle\rangle \qquad \text{(by 2.)} \\
&\quad \cong [f|c]\langle s_1 \cdot \langle \mathcal{L}_1 \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle, \dots , \mathcal{L}_n \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle\rangle, \dots , \\
&\qquad\qquad s_k \cdot \langle \mathcal{L}_1 \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle, \dots , \mathcal{L}_n \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle\rangle \\
&\quad \cong [f|c]\langle s_1, \dots , s_k\rangle \cdot \langle \mathcal{L}_1 \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle, \dots , \mathcal{L}_n \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle\rangle\rangle
\end{aligned}$$

Finally we compute

$$\begin{aligned}
&(\mathcal{L} \cdot \langle \mathcal{L}_1, \dots , \mathcal{L}_n\rangle) \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle \\
&\qquad = \left(\coprod_{t\in\mathcal{L}} |t| \cdot \langle \mathcal{L}_1, \dots , \mathcal{L}_n\rangle\right) \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle \\
&\qquad\quad \text{and since OI-substitution preserves colimits on the left:} \\
&\qquad \cong \coprod_{t\in\mathcal{L}} ((|t| \cdot \langle \mathcal{L}_1, \dots , \mathcal{L}_n\rangle) \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle) \\
&\qquad \cong \coprod_{t\in\mathcal{L}} |t| \cdot \langle \mathcal{L}_1 \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle, \dots , \mathcal{L}_n \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle\rangle \\
&\qquad = \mathcal{L} \cdot \langle \mathcal{L}_1\langle \mathcal{M}_1, \dots , \mathcal{M}_n\rangle, \dots , \mathcal{L}_n \cdot \langle \mathcal{M}_1, \dots , \mathcal{M}_m\rangle\rangle.
\end{aligned}$$

□

9.6 Lemma. *Let $\mathcal{L} \in \mathit{WTL}_{\Sigma(X_n)}$, $\mathcal{L}_1, \dots , \mathcal{L}_n \in \mathit{WTL}_{\Sigma(X_m)}$ and let $t \in \mathrm{WT}_{\Sigma(X_n)}$. Then*

1. $t \cdot \langle \mathcal{L}_1, \dots , \mathcal{L}_n\rangle \cong (t \cdot_{x_1} \{[x_{m+1}|1]\} \cdots \cdot_{x_n} \{[x_{m+n}|1]\}) \cdot_{x_{m+1}} \mathcal{L}_1 \cdots \cdot_{x_{m+n}} \mathcal{L}_n,$
2. $\mathcal{L} \cdot \langle \mathcal{L}_1, \dots , \mathcal{L}_n\rangle \cong (\mathcal{L} \cdot_{x_1} \{[x_{m+1}|1]\} \cdots \cdot_{x_n} \{[x_{m+n}|1]\}) \cdot_{x_{m+1}} \mathcal{L}_1 \cdots \cdot_{x_{m+n}} \mathcal{L}_n.$

Proof. **about 1:** We proceed by induction on the structure of t.

$$\begin{aligned}
[x_i|c] \cdot \langle \mathcal{L}_1, \dots , \mathcal{L}_n\rangle &= c \cdot \mathcal{L}_i \\
&= [x_{m+i}|c] \cdot_{x_{m+1}} \mathcal{L}_1 \cdots \cdot_{x_{m+n}} \mathcal{L}_n \\
&= ([x_i|c] \cdot_{x_1} \{[x_{m+1}|1]\} \cdots \cdot_{x_n} \{[x_{m+n}|1]\}) \cdot_{m+1} \mathcal{L}_1 \cdots \cdot_{x_{m+n}} \mathcal{L}_n.
\end{aligned}$$

$$\begin{aligned}
[a|c] \cdot \langle \mathcal{L}_1, \dots , \mathcal{L}_n\rangle &= \{[a|c]\} \\
&= \{[a|c]\} \cdot_{x_{m+1}} \mathcal{L}_1 \cdots \cdot_{x_{m+n}} \mathcal{L}_n \\
&= ([a|c] \cdot_{x_1} \{[x_{m+1}]\} \cdots \cdot_{x_n} \{[x_{m+n}|1]\}) \cdot_{x_{m+1}} \mathcal{L}_1 \cdots \cdot_{x_{m+n}} \mathcal{L}_n.
\end{aligned}$$

$$
\begin{aligned}
&([f|c]\langle s_1,\dots,s_k\rangle)\cdot\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle = [f|c]\langle s_1\cdot\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle,\dots,s_k\cdot\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle\rangle\\
&\quad = [f|c]\langle (s_i\cdot_{x_1}\{[x_{m+1}|1]\}\cdots\cdot_{x_n}\{[x_{m+n}|1]\})\cdot_{x_{m+1}}\mathcal{L}_1\cdots\cdot_{x_{m+n}}\mathcal{L}_n\rangle_{i=1}^{k}\\
&\qquad \text{and by 2.18:}\\
&\quad \cong [f|c]\langle s_i\cdot_{x_1}\{[x_{m+1}|1]\}\cdots\cdot_{x_n}\{[x_{m+n}|1]\}\rangle_{i=1}^{n}\cdot_{x_{m+1}}\mathcal{L}_1\cdots\cdot_{x_{m+n}}\mathcal{L}_n\\
&\qquad \text{and by 2.18 and 2.10:}\\
&\quad \cong ([f|c]\langle s_1,\dots,s_k\rangle\cdot_{x_1}\{[x_{m+1}|1]\}\cdots\cdot_{x_n}\{[x_{m+n}|1]\})\cdot_{x_{m+1}}\mathcal{L}_1\cdots\cdot_{x_{m+n}}\mathcal{L}_n.
\end{aligned}
$$

about 2: This follows from part 1 and from the fact that OI-substitution preserves colimits on the left. □

9.7 A theory of weighted tree-languages. Next we will identify an interesting subtheory of $\mathrm{Th}_\omega(\mathsf{WTL}_{\Sigma(X)})$. Given a tuple $(\mathcal{L}_1,\dots,\mathcal{L}_m)$ of weighted tree-languages from $\mathsf{WTL}_{\Sigma(X_n)}$, we define a functor $F_{\langle\mathcal{L}_1,\dots,\mathcal{L}_m\rangle} : \mathsf{WTL}^n_{\Sigma(X)} \longrightarrow \mathsf{WTL}^m_{\Sigma(X)}$ according to

$$(\mathcal{M}_1,\dots,\mathcal{M}_n)\mapsto(\mathcal{L}_1\cdot\langle\mathcal{M}_1,\dots,\mathcal{M}_n\rangle,\dots,\mathcal{L}_m\cdot\langle\mathcal{M}_1,\dots,\mathcal{M}_n\rangle).$$

By 9.4 $F_{\langle\mathcal{L}_1,\dots,\mathcal{L}_m\rangle}$ is indeed a functor that preserves directed colimits. Hence it is an arrow from $m\longrightarrow n$ in $\mathrm{Th}_\omega(\mathsf{WTL}_{\Sigma(X)})$. With $\mathrm{WTh}_{\Sigma(X)}$ we denote the smallest sub iteration theory of $\mathrm{Th}_\omega(\mathsf{WTL}_{\Sigma(X)})$ that contains all functors $\{F_{\langle\mathcal{L}_1,\dots,\mathcal{L}_m\rangle} \mid \mathcal{L}_1,\dots,\mathcal{L}_m\in\mathsf{WTL}_{\Sigma(X_n)},\ n,m\in\mathbb{N}\}$.

It is important to note that the assignment $(\mathcal{L}_1,\dots,\mathcal{L}_m)\mapsto F_{\langle\mathcal{L}_1,\dots,\mathcal{L}_m\rangle}$ is reversible up to isomorphism since

$$F_{\langle\mathcal{L}_1,\dots,\mathcal{L}_m\rangle}(\{[x_1|1]\},\dots,\{[x_n|1]\})\cong(\mathcal{L}_1,\dots,\mathcal{L}_m)=:\mathcal{L}(F_{\langle\mathcal{L}_1,\dots,\mathcal{L}_m\rangle})$$

where the isomorphism of tuples of weighted tree-languages is defined component wise. Instead of $F_{\langle\mathcal{L}\rangle}$ we will usually write just $F_{\mathcal{L}}$.

If we are given two functors $F_1,F_2 : m\longrightarrow n$ from $\mathrm{Th}_\omega(\mathsf{WTL}_{\Sigma(X)})$ such that for all $\mathcal{M}_1,\dots,\mathcal{M}_n\in\mathsf{WTL}_{\Sigma(X)}$ we have $F_1(\mathcal{M}_1,\dots,\mathcal{M}_n)\cong F_2(\mathcal{M}_1,\dots,\mathcal{M}_n)$, then we will abbreviate this situation by $F_1\equiv F_2$ and call the two functors equivalent. However, do not confuse this with the stronger notion of natural isomorphism.

9.8 Lemma. *All morphisms of* $\mathrm{WTh}_{\Sigma(X)}$ *preserve monos.*

Proof. All generators of $\mathrm{WTh}_{\Sigma(X)}$ preserve monos. The tupling and the composition of monos preserving functors preserves monos.

If $F : n\longrightarrow n+p$ preserves monos, then the claim that $F^\dagger$ preserves monos follows immediately from the definition of $F^\dagger$ (cf. the diagram in 8.22). □

9.9 Remark. Let $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X_{1+p})}$. Then $F_{\mathcal{L}} : 1 \longrightarrow 1+p$ and hence $F_{\mathcal{L}}^{\dagger} : 1 \longrightarrow p$ in $\mathrm{Th}_{\omega}(\mathsf{WTL}_{\Sigma(X)})$. Recall that the value $F_{\mathcal{L}}^{\dagger}(\mathcal{M}_1, \dots, \mathcal{M}_p)$ is the initial algebra carrier of the functor $F := F_{\mathcal{L}}(-, \mathcal{M}_1, \dots, \mathcal{M}_p) : \mathsf{WTL}_{\Sigma(X)} \longrightarrow \mathsf{WTL}_{\Sigma(X)}$ which in our case may be obtained as the colimit of the initial F-cochain

$$\emptyset \xrightarrow{\;!\;} F(\emptyset) \xrightarrow{F(!)} F^2(\emptyset) \xrightarrow{F^2(!)} \cdots$$

Note that by 2.21, given two morphisms $f_1, f_2 : 1 \longrightarrow 1+p$ of $\mathrm{Th}_{\omega}(\mathsf{WTL}_{\Sigma(X)})$ preserving monos, we have $f_1 \equiv f_2 \Rightarrow f_1^{\dagger} \equiv f_2^{\dagger}$.

In the following we will make a connection between the dagger-operation on $F_{\mathcal{L}}$ and the x_1-recursion of $\mathcal{L}$:

9.10 Lemma. *With the notions from above*

$$\forall \mathcal{M}_1, \dots, \mathcal{M}_p \in \mathit{WTL}_{\Sigma(X)} : F_{\mathcal{L}}^{\dagger}(\mathcal{M}_1, \dots, \mathcal{M}_p) \cong F_{\mathcal{L}^{\mu}_{x_1}}(\{[x_1|1]\}, \mathcal{M}_1, \dots, \mathcal{M}_p).$$

Proof. $F_{\mathcal{L}}^{\dagger}(\mathcal{M}_1, \dots, \mathcal{M}_p)$ is the colimit of the initial cochain

$$\emptyset \rightarrowtail F_{\mathcal{L}}(\emptyset, \mathcal{M}_1, \dots, \mathcal{M}_p) \rightarrowtail F_{\mathcal{L}}^2(\emptyset, \mathcal{M}_1, \dots, \mathcal{M}_p) \rightarrowtail \cdots$$

by 2.21 we do not need to worry about the morphisms in the cochain as long as they are injective. Therefore we write (slightly abusing notation) that

$$F_{\mathcal{L}}^{\dagger}(\mathcal{M}_1, \dots, \mathcal{M}_p) = \operatorname*{colim}_{n \to \infty} F_{\mathcal{L}}^n(\emptyset, \mathcal{M}_1, \dots, \mathcal{M}_p).$$

We will show that the functors

$$F_{\mathcal{L}}(-, \mathcal{M}_1, \dots, \mathcal{M}_p) \text{ and } F_{\mathcal{L}}(-, \{[x_2|1]\}, \dots, \{[x_{p+1}|1]\}) \cdot \langle \{[x_1|1]\}, \mathcal{M}_1, \dots, \mathcal{M}_p \rangle$$

have isomorphic initial cochains. Of course this is done inductively. Both cochains start with $\emptyset$ which sets the induction-base. Suppose we already showed

$$F_{\mathcal{L}}^n(\emptyset, \mathcal{M}_1, \dots, \mathcal{M}_p) \cong F_{\mathcal{L}}^n(\emptyset, \{[x_2|1]\}, \dots, \{[x_{1+p}|1]\}) \cdot \langle \{[x_1|1]\}, \mathcal{M}_1, \dots, \mathcal{M}_p \rangle.$$

Then

$$\begin{aligned}
F_{\mathcal{L}}^{n+1}(\emptyset, \mathcal{M}_1, \dots, \mathcal{M}_p) &= F_{\mathcal{L}}(F_{\mathcal{L}}^n(\emptyset, \mathcal{M}_1, \dots, \mathcal{M}_p), \mathcal{M}_1, \dots, \mathcal{M}_p) \\
&= \mathcal{L} \cdot \langle F_{\mathcal{L}}^n(\emptyset, \mathcal{M}_1, \dots, \mathcal{M}_p), \mathcal{M}_1, \dots, \mathcal{M}_p \rangle \\
&\cong \mathcal{L} \cdot \langle F_{\mathcal{L}}^n(\emptyset, \{[x_1|1]\}, \dots, \{[x_{p+1}|1]\}) \\
&\quad \cdot \langle \{[x_1|1]\}, \mathcal{M}_1, \dots, \mathcal{M}_p \rangle, \mathcal{M}_1, \dots, \mathcal{M}_p \rangle
\end{aligned}$$

with $\mathcal{M}_i = \{[x_{i+1}|1]\} \cdot \langle \{[x_1|1]\}, \mathcal{M}_1, \dots, \mathcal{M}_p \rangle$ and 9.5(3):

$$\begin{aligned}
&\cong \mathcal{L} \cdot \langle F_{\mathcal{L}}^n(\emptyset, \{[x_2|1]\}, \dots, \{[x_{p+1}|1]\}), \{[x_2|1]\}, \dots, \{[x_{p+1}|1]\} \rangle \\
&\quad \cdot \langle \{[x_1|1]\}, \mathcal{M}_1, \dots, \mathcal{M}_p \rangle \\
&= F_{\mathcal{L}}^{n+1}(\emptyset, \{[x_2|1]\}, \dots, \{[x_{p+1}|1]\}) \cdot \langle \{[x_1|1]\}, \mathcal{M}_1, \dots, \mathcal{M}_p \rangle.
\end{aligned}$$

Now we use, that OI-substitution preserves colimits on the left (cf. 9.4) and obtain

$$\begin{aligned}&\operatorname*{colim}_{n\to\infty}(F^n_{\mathcal{L}}(\emptyset,\{[x_2|1]\},\dots,\{[x_{p+1}|1]\})\cdot\langle\{[x_1|1]\},\mathcal{M}_1,\dots,\mathcal{M}_p\rangle)\\ &\qquad\cong(\operatorname*{colim}_{n\to\infty}F^n_{\mathcal{L}}(\emptyset,\{[x_2|1]\},\dots,\{[x_{p+1}|1]\}))\cdot\langle\{[x_1|1]\},\mathcal{M}_1,\dots,\mathcal{M}_p\rangle.\end{aligned}$$

But $F_{\mathcal{L}}(-,\{[x_2|1]\},\dots,\{[x_{p+1}|1]\})\cong(\mathcal{L}\cdot_{x_1}-)=R_{x_1}$ (cf. 2.26). Hence

$$\operatorname*{colim}_{n\to\infty}F^n_{\mathcal{L}}(\emptyset,\{[x_2|1]\},\dots,\{[x_{p+1}|1]\})\cong\operatorname*{colim}_{n\to\infty}R^n_{x_1}(\emptyset,\mathcal{L})=\mathcal{L}^\mu_{x_1}.$$

Summing up, we have

$$\begin{aligned}F^\dagger(\mathcal{M}_1,\dots,\mathcal{M}_p)&=\operatorname*{colim}_{n\to\infty}F^n_{\mathcal{L}}(\emptyset,\mathcal{M}_1,\dots,\mathcal{M}_p)\\&\cong\mathcal{L}^\mu_{x_1}\cdot\langle\{[x_1|1]\},\mathcal{M}_1,\dots,\mathcal{M}_p\rangle\\&=F_{\mathcal{L}^\mu_{x_1}}(\{[x_1|1]\},\mathcal{M}_1,\dots,\mathcal{M}_p).\end{aligned}$$

□

9.11 Remark. The previous Lemma together with 9.5(3) sheds some light onto the iteration theory $\mathrm{WTh}_{\Sigma(X)}\leq\mathrm{Th}_\omega(\mathsf{WTL}_{\Sigma(X)})$. First we notice that for each $\mathcal{L}\in\mathsf{WTL}_{\Sigma(X_{1+p})}$ there exists an $\mathcal{L}'\in\mathsf{WTL}_{\Sigma(X_p)}$ such that $F^\dagger_{\mathcal{L}}\equiv F_{\mathcal{L}'}$. In particular we may chose $\mathcal{L}':=\mathcal{L}^\mu_{x_1}\cdot\langle\{[x_1|1]\},\{[x_1|1]\},\dots,\{[x_p|1]\}\rangle$. But from this it follows also that $F^\dagger_{\mathcal{L}}\equiv F_{\mathcal{L}^\mu_{x_1}}\cdot\langle 1_p,\mathbf{1}_p\rangle$. In the following proposition we will show that a similar result holds for all morphisms of $\mathrm{WTh}_{\Sigma(X)}$.

9.12 Proposition. *For each $F\,:n\longrightarrow m$ in $\mathrm{WTh}_{\Sigma(X)}$ there are weighted tree-languages $\mathcal{L}_1,\dots,\mathcal{L}_n\in\mathsf{WTL}_{\Sigma(X_m)}$ such that $F\equiv F_{\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle}$.*

Proof. First we argue that the morphisms of $\mathrm{WTh}_{\Sigma(X)}$ with the claimed property form a subtheory. That is, they contain the distinguished morphisms and are closed with respect to composition and source-tupling. Clearly $i_n=F_{\{[x_i|1]\}}$ and $F_{\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle}=\langle F_{\mathcal{L}_1},\dots,F_{\mathcal{L}_n}\rangle$ by definition. Let $F\,:\,n\longrightarrow m$, $G\,:\,m\longrightarrow p$ in $\mathrm{WTh}_{\Sigma(X)}$ such that $F\equiv F_{\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle}$, $G\equiv F_{\langle\mathcal{L}'_1,\dots,\mathcal{L}'_m\rangle}$. Then $F\cdot G\,:\,n\longrightarrow p$ and for all $\mathcal{M}_1,\dots,\mathcal{M}_p\in\mathsf{WTL}_{\Sigma(X)}$:

$$\begin{aligned}(F\cdot G)(\mathcal{M}_1,\dots,\mathcal{M}_p)&\cong F_{\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle}(F_{\langle\mathcal{L}'_1,\dots,\mathcal{L}'_m\rangle}(\mathcal{M}_1,\dots,\mathcal{M}_p))\\&=\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle\cdot(\langle\mathcal{L}'_1,\dots,\mathcal{L}'_m\rangle\cdot\langle\mathcal{M}_1,\dots,\mathcal{M}_p\rangle)\\&=\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle\cdot(\langle\mathcal{L}'_j\cdot\langle\mathcal{M}_1,\dots,\mathcal{M}_p\rangle\rangle_{j=1}^m)\\&=\langle\mathcal{L}_i\cdot\langle\mathcal{L}_j\cdot\langle\mathcal{M}_1,\dots,\mathcal{M}_p\rangle\rangle_{j=1}^m\rangle_{i=1}^n\end{aligned}$$

by 9.5(3):

$$\begin{aligned}&\cong\langle\mathcal{L}_i\cdot\langle\mathcal{L}'_1,\dots,\mathcal{L}'_m\rangle\rangle_{i=1}^n\cdot\langle\mathcal{M}_1,\dots,\mathcal{M}_p\rangle\\&=F_{\langle\mathcal{L}_i\cdot\langle\mathcal{L}'_1,\dots,\mathcal{L}'_m\rangle\rangle_{i=1}^n}(\mathcal{M}_1,\dots,\mathcal{M}_p).\end{aligned}$$

It remains to show that application of dagger preserves the claimed property. By 8.13 it is enough to show this for scalar morphisms. Let $F : 1 \longrightarrow 1+p$ such that $F \equiv F_{\mathcal{L}}$ for some $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X_{1+p})}$. Then by 9.10

$$\begin{aligned} F^\dagger &\equiv F_{\mathcal{L}}^\dagger \\ &\equiv F_{\mathcal{L}^\mu_{x_1}\cdot\langle 1_p,1_p,2_p,\dots,p_p\rangle} \\ &\equiv F_{\mathcal{L}^\mu_{x_1}\cdot\langle\{[x_1|1]\},\{[x_1|1]\},\dots,\{[x_p|1]\}\rangle} \qquad \text{(cf. 9.11)} \end{aligned}$$

□

9.13 Remark. Note that in the previous proof we showed more than the mere existence of $\mathcal{L}(F)$ but also for instance that for $F : n \longrightarrow m$, $G : m \longrightarrow p$ from $\mathrm{WTh}_{\Sigma(X)}$ we have $\mathcal{L}(F\cdot G) \cong \mathcal{L}(F)\cdot\mathcal{L}(G)$ where

$$\begin{aligned} \mathcal{L}(F)\cdot\mathcal{L}(G) &= \langle \mathcal{L}_1(F),\dots,\mathcal{L}_n(F)\rangle\cdot\langle\mathcal{L}_1(G),\dots,\mathcal{L}_m(G)\rangle \\ &:= \langle \mathcal{L}_i(F)\cdot\langle\mathcal{L}_1(G),\dots,\mathcal{L}_m(G)\rangle\rangle_{i=1}^n. \end{aligned}$$

It is easy to see now that $\equiv$ is in fact a T-congruence of $\mathrm{WTh}_{\Sigma(X)}$.

9.14 Finitarity, quasiregularity and (weak) recognizability on $\mathbf{WTh}_{\Sigma(X)}$. The last proposition allows us in principle to identify each arrow $F : n \longrightarrow m$ of $\mathrm{WTh}_{\Sigma(X)}$ with a tuple $\mathcal{L}(F) = \langle\mathcal{L}_1(F),\dots,\mathcal{L}_n(F)\rangle$ of weighted tree-languages from $\mathsf{WTL}_{\Sigma(X_m)}$. Because of this identification we may associate attributes of weighted tree-languages such as (weak) recognizability, quasiregularity or finitarity to morphisms of $\mathrm{WTh}_{\Sigma(X)}$.

An arrow $F : n \longrightarrow m$ of $\mathrm{WTh}_{\Sigma(X)}$ is called *(weakly) recognizable* if $\mathcal{L}_i(F)$ is weakly recognizable for all $i = 1,\dots,n$. It is called *finitary* if $\mathcal{L}_i(F)$ is finitary for every $i = 1,\dots,n$.

An arrow $F : n \longrightarrow n+p$ of $\mathrm{WTh}_{\Sigma(X)}$ is called *quasiregular* if $\mathcal{L}_i(F)$ is x_j-quasiregular for $i,j = 1,\dots,n$.

9.15 Remark. By 9.6, and by 2.33 the OI-substitution preserves finitarity. Hence the finitary arrows of $\mathrm{WTh}_{\Sigma(X)}$ form a subtheory. This subtheory shall be denoted by $\mathrm{WTh}^{\mathrm{fin}}_{\Sigma(X)}$.

9.16 Proposition. *Let $F : n \longrightarrow n+p$ be from $\mathrm{WTh}_{\Sigma(X)}$. Then*

$$\mathcal{L}_i(F^\dagger) \in \mathrm{wRat}(\mathcal{L}_1(F),\dots,\mathcal{L}_n(F)) \quad (i = 1,\dots,n).$$

Moreover, if F is quasiregular and finitary, then

$$\mathcal{L}_i(F^\dagger) \in \mathrm{Rat}(\mathcal{L}_1(F),\dots,\mathcal{L}_n(F)) \quad (i = 1,\dots,n).$$

In particular $\mathcal{L}(F^\dagger)$ is finitary. If $G \equiv F$, then $G^\dagger \equiv F^\dagger$.

Proof. We proceed the same way as in 8.13. That is, we do an induction on n and we use the pairing identity.

If $F, G : 1 \longrightarrow 1+p$ then

$$\begin{aligned}\mathcal{L}(F^\dagger) &\cong \mathcal{L}_1(F)^\mu_{x_1} \cdot \langle \{[x_1|1]\}, \{[x_1|1]\}, \dots, \{[x_p|1]\}\rangle \\ &\cong \mathcal{L}_1(F)^\mu_{x_1} \cdot_{x_1} \{[x_1|1]\} \cdot_{x_2} \{[x_1|1]\} \cdots \cdot_{x_{1+p}} \{[x_p|1]\}.\end{aligned}$$

In particular $\mathcal{L}_1(F^\dagger) \in \mathrm{wRat}(\mathcal{L}_1(F))$. Since $\mathcal{L}_1(F) \cong \mathcal{L}_1(G)$, we have $\mathcal{L}_1(F)^\mu_{x_1} \cong \mathcal{L}_1(G)^\mu_{x_1}$ and hence $\mathcal{L}(F^\dagger) \cong \mathcal{L}(G^\dagger)$ which is equivalent to $G^\dagger \equiv F^\dagger$.

By 2.34 and 2.27, if $\mathcal{L}_1(F)$ is finitary and x_1-quasiregular, then $\mathcal{L}_1(F)^\mu_{x_1}$ is finitary. Then 2.33(4) gives that $\mathcal{L}_1(F)^\mu_{x_1} \cdot_{x_1} \{[x_1|1]\} \cdots \cdot_{x_{1+p}} \{[x_p|1]\}$ is finitary. In particular $\mathcal{L}_1(F^\dagger) \in \mathrm{Rat}(\mathcal{L}_1(F))$.

If $F : n+1 \longrightarrow n+1+p$, then $F = \langle \hat{F}, F'\rangle$ where $\hat{F} : 1 \longrightarrow n+1+p$ and $F' : n \longrightarrow n+1+p$. Hence, by the pairing identity, $F^\dagger \cong \langle \hat{F}^\dagger \cdot \langle H_F^\dagger, \mathbf{1}_p\rangle, H_F^\dagger\rangle$ where $H_F = F' \cdot \langle \hat{F}^\dagger, \mathbf{1}_{n+p}\rangle : n \longrightarrow n+p$. By 9.14 we have

$$\begin{aligned}\mathcal{L}(H_F) &\cong \mathcal{L}(F') \cdot \langle \mathcal{L}(\hat{F}^\dagger), \mathcal{L}(\mathbf{1}_{n+p})\rangle \\ &= \langle \mathcal{L}_1(F'), \dots, \mathcal{L}_n(F')\rangle \cdot \langle \mathcal{L}_1(\hat{F}^\dagger), \{[x_1|1]\}, \dots, \{[x_{n+p}|1]\}\rangle\end{aligned}$$

and by 9.11(3)

$$\cong \langle \mathcal{L}_i(F') \cdot \langle \mathcal{L}_1(\hat{F}^\dagger), \{[x_1|1]\}, \dots, \{[x_{n+p}|1]\}\rangle\rangle_{i=1}^n$$

Since $\mathcal{L}_i(F') = \mathcal{L}_{i+1}(F)$ and since by the induction hypothesis $\mathcal{L}_1(\hat{F}^\dagger)$ is contained in $\mathrm{wRat}(\mathcal{L}_1(F), \dots, \mathcal{L}_{n+1}(F))$, we conclude $\mathcal{L}_i(H_F) \in \mathrm{wRat}(\mathcal{L}_1(F), \dots, \mathcal{L}_{n+1}(F))$ $(i = 1, \dots, n)$. By induction hypothesis $\mathcal{L}_i(H_F^\dagger) \in \mathrm{wRat}(\mathcal{L}_1(H_F), \dots, \mathcal{L}_n(H_F))$ $(i = 1, \dots, n)$. Hence we obtain that $\mathcal{L}_i(H_F^\dagger) \in \mathrm{wRat}(\mathcal{L}_1(F), \dots, \mathcal{L}_{n+1}(F))$ $(i = 1, \dots, n)$.

Now, if F is finitary and quasiregular, then $\mathcal{L}_1(\hat{F}^\dagger) \cong \mathcal{L}_1(\hat{F})^\mu_{x_1} \cdot_{x_1} \{[x_1|1]\} \cdots \cdot_{x_{n+1+p}} \{[x_{n+p}|1]\}$ is finitary and x_i-quasiregular for $i = 1, \dots, n$. Hence H_F is finitary and quasiregular. By induction hypothesis $\mathcal{L}_i(H_F^\dagger)$ is in $\mathrm{Rat}(\mathcal{L}_1(H_F), \dots, \mathcal{L}_n(H_F))$ and is finitary $(i = 1, \dots, n)$. Again by induction hypothesis $\mathcal{L}(\hat{F}^\dagger)$ is finitary and $\mathcal{L}_1(\hat{F}^\dagger) \in \mathrm{Rat}(\mathcal{L}_1(\hat{F}))$, we finally conclude that $\mathcal{L}(F^\dagger)$ is finitary and that $\mathcal{L}_i(F^\dagger)$ is contained in $\mathrm{Rat}(\mathcal{L}_1(F), \dots, \mathcal{L}_{n+1}(F))$ $(i = 1, \dots, n+1)$.

If $G \equiv F$, then $G = \langle \hat{G}, G'\rangle$ with $\hat{G} : 1 \longrightarrow n+1+p$, $G' : n \longrightarrow n+1+p$ and $\hat{G} \equiv \hat{F}$, $G' \equiv F'$. Again by the pairing-identity $G^\dagger \cong \langle \hat{G}^\dagger \cdot \langle H_G^\dagger, \mathbf{1}_p\rangle, H_g^\dagger\rangle$ where $H_G = G' \cdot \langle \hat{G}^\dagger, \mathbf{1}_{n+p}\rangle : n \longrightarrow n+p$. By induction-hypothesis $\hat{G}^\dagger \equiv \hat{F}^\dagger$. By 9.14, the equivalence relation $\equiv$ is a T-congruence. Hence $H_F \equiv H_G$. Again by induction hypothesis $H_F^\dagger \equiv H_G^\dagger$ and finally $G^\dagger \equiv F^\dagger$. □

9.17 Corollary. *The weakly recognizable arrows of* $\mathrm{WTh}_{\Sigma(X)}$ *form a sub iteration theory of* $\mathrm{WTh}_{\Sigma(X)}$.

Proof. By 9.6 and by 4.23 the weakly recognizable weighted tree-languages are closed with respect to OI-substitution. By 9.13 the weakly recognizable arrows are closed with respect to composition. By 9.16 the weakly recognizable arrows are also closed with respect to the dagger-operation. This completes the proof. □

9.18 Normal descriptions. A weighted tree t is called *primitive* if it is of the shape $t = [f|c]\langle[x_{i_1}|1], \dots, [x_{i_n}|1]\rangle$ for some $f \in \Sigma(X)^{(n)}$, $n \in \mathbb{N}$, $c \in K$, $x_{i_j} \in X$, $j = 1, \dots, n$.

A weighted tree-language $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X)}$ is called *primitive* if it is finite and each of its elements is primitive.

A morphism $F : q \longrightarrow n + p$ from $\mathrm{WTh}_{\Sigma(X)}$ is called *primitive* of weight n if $\mathcal{L}_i(F)$ is primitive for $i = 1, \dots, q$.

A *normal description* $k \longrightarrow p$ of weight n is a pair $D = (\alpha, F)$ such that $F : n \longrightarrow n + p$ is primitive of weight n and $\alpha : k \longrightarrow n$ is base. The behavior of $D : k \longrightarrow p$ is the morphism $|D| := \alpha \cdot F^{\dagger}$. D is called quasiregular if F is quasiregular.

9.19 Proposition. *For each normal description $D : k \longrightarrow p$, $|D|$ is weakly recognizable. If D is quasiregular, then $|D|$ is recognizable.*

Proof. Every finite weighted tree-language is recognizable. Hence every primitive morphism is recognizable. The rest follows from 9.16. □

9.20 Proposition. *Let $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X_p)}$. If $\mathcal{L}$ is weakly recognizable, then there exists a normal description $D : 1 \longrightarrow p$ such that $F_{\mathcal{L}} \equiv |D|$. If $\mathcal{L}$ is recognizable, then D may be chosen quasiregular.*

Proof. Suppose $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X_p)}$ is weakly recognizable. Let $\mathcal{A} = (Q, q_1, T, \lambda, S, \sigma)$ be a wWTA with $Q = \{q_1, \dots, q_n\}$ such that $\mathcal{L}_{\mathcal{A}} \cong \mathcal{L}$.

Let $T_i := \{\tau \in T \mid \mathrm{dom}(\tau) = q_i\}$, $S_i := \{s \in S \mid \mathrm{dom}(s) = q_i\}$, $M_i = T_i \cup S_i$, and for $t \in M_i$ define

$$|t|_i := \begin{cases} [f|c]\langle[x_{i_1}|1], \dots, [x_{i_k}|1]\rangle & \text{if } t \in T_i, \lambda(t) = (q_i, f, q_{i_1}, \dots, q_{i_k}, c), f \notin X \\ [x_{j+n}|c] & \text{if } t \in T_i, \lambda(t) = (q_i, x_j, c) \\ [x_j|c] & \text{if } t \in S_i, \sigma(s) = (q_i, c, q_j). \end{cases}$$

Then $\mathcal{M}_i = (M_i, |.|_i)$ is primitive. Hence $F := \langle F_{\mathcal{M}_1}, \dots, F_{\mathcal{M}_n}\rangle : n \longrightarrow n + p$ is primitive. Set $\alpha := F_{\{[x_1|1]\}}$. Then $D := (\alpha, F)$ is a normal description. We will show that $\mathcal{L} \cong |D|$.

Let L_i be the set of all runs of $\mathcal{A}$ with root q_i. Recall that L_i may be obtained inductively (cf. 4.3). Together with the structure map that was defined in 4.3, $\mathcal{L}_i = (L_i, |.|) \in \mathsf{WTL}_{\Sigma(X_p)}$.

For $i = 1, \dots, n$ define $L_{i,0} := \emptyset$ and

$$\begin{aligned} L_{i,m+1} := \{\tau\langle r_1, \dots, r_k\rangle \mid \tau \in T_i,\ \lambda(\tau) = (q_i, f, q_{i_1}, \dots, q_{i_k}, c),\ r_j \in L_{i_j,m}\} \\ \cup \{s \cdot r \mid s \in S_i,\ \sigma(s) = (q_i, c, q_j),\ r \in \mathcal{L}_{j,m}\}. \end{aligned}$$

Then $\bigcup_{j \in \mathbb{N}} L_{i,j} = L_i$. Moreover, with $\mathcal{L}_{i,j} = (L_{i,j}, |.|)$ we have $\mathcal{L}_{i,j} \leq \mathcal{L}_{i,j+1}$ ($i = 1, \dots, n$, $j \in \mathbb{N}$). With $\varphi_{i,j} : \mathcal{L}_{i,j} \rightarrowtail \mathcal{L}_{i,j+1}$ being the identical embedding we get that $(\mathcal{L}_{i,j}, \varphi_{i,j})_{j \in \mathbb{N}}$ is an injective ω-cochain such that $\mathcal{L}_i \cong \operatorname*{colim}_{j \to \infty} \mathcal{L}_{i,j}$.

On the other hand consider the functor $\hat{F} = F(-,\dots,-,\{[x_1|1]\},\dots,\{[x_p|1]\})$. Then $\mathcal{L}(F^\dagger) = F^\dagger(\{[x_1|1]\},\dots,\{[x_p|1]\})$ is the initial algebra carrier of $\hat{F}$. Since $\hat{F}$ preserves directed colimits, this may be obtained as the colimit of the initial cochain of $\hat{F}$.

We will show that the initial sequence of $\hat{F}$ is isomorphic to the injective ω-cochain $((\mathcal{L}_{i,j})_{i=1}^n, (\varphi_{i,j})_{i=1}^n)_{j\in\mathbb{N}}$. Let $(\hat{F}^n(\emptyset,\dots,\emptyset), \hat{F}^n(!,\dots,!))_{n\in\mathbb{N}}$ be the initial ω-cochain of $\hat{F}$. Then $\hat{F}^0(\emptyset,\dots,\emptyset) = (\emptyset,\dots,\emptyset) = (\mathcal{L}_{1,0},\dots,\mathcal{L}_{n,0})$ and thus the induction anchor is set. Suppose we showed that $\hat{F}^m(\emptyset,\dots,\emptyset) = (\hat{\mathcal{L}}_{1,m},\dots,\hat{\mathcal{L}}_{n,m}) \cong (\mathcal{L}_{1,m},\dots,\mathcal{L}_{n,m})$. Then

$$\begin{aligned}
\hat{\mathcal{L}}_{i,m+1} &= \hat{F}(\hat{\mathcal{L}}_{1,m},\dots,\hat{\mathcal{L}}_{n,m})_i \cong \hat{F}(\mathcal{L}_{1,m},\dots,\mathcal{L}_{n,m})_i \\
&= F(\mathcal{L}_{1,m},\dots,\mathcal{L}_{n,m},\{[x_1|1]\},\dots,\{[x_p|1]\})_i \\
&= \mathcal{M}_i \cdot \langle \mathcal{L}_{1,m},\dots,\mathcal{L}_{n,m},\{[x_1|1]\},\dots,\{[x_p|1]\}\rangle \\
&= \coprod_{t\in\mathcal{M}_i} |t|_i \cdot \langle \mathcal{L}_{1,m},\dots,\mathcal{L}_{n,m},\{[x_1|1]\},\dots,\{[x_p|1]\}\rangle.
\end{aligned}$$

Now we notice that

$$\begin{aligned}
&|t|_i \cdot \langle \mathcal{L}_{1,m},\dots,\mathcal{L}_{n,m},\{[x_1|1]\},\dots,\{[x_p|1]\}\rangle \\
&\qquad\cong \begin{cases} [f|c]\langle \mathcal{L}_{i_1,m},\dots,\mathcal{L}_{i_k,m}\rangle & \text{if } |t|_i = [f|c]\langle [x_{i_1}|1],\dots,[x_{i_k}|1]\rangle \\ c\cdot \mathcal{L}_{j,m} & \text{if } |t|_i = [x_j|c] \\ \{[x_j|c]\} & \text{if } |t|_i = [x_{j+n}|c]. \end{cases}
\end{aligned}$$

Let us denote this language by $\hat{\mathcal{N}_t}$. On the other hand we may write

$$\mathcal{L}_{i,m+1} = \coprod_{t\in M_i} \mathcal{N}_t$$

where

$$\mathcal{N}_t = (\{t\langle r_1,\dots,r_k\rangle \mid \lambda(t) = (q_i,f,q_{i_1},\dots,q_{i_k},c),\ r_j \in \mathcal{L}_{i_j,m}\}, |.|) \qquad \text{if } t\in T_i$$

and

$$\mathcal{N}_t = (\{s\cdot r \mid \sigma(s) = (q_i,c,q_j),\ r\in L_{j,m}\}, |.|) \qquad \text{if } t\in S_i.$$

If $t\in T_i$ and $\operatorname{lab}(t)\notin X$, then it is obvious that $\mathcal{N}_t \cong \hat{\mathcal{N}_t}$.

If $t\in T_i$ and $\lambda(t) = (q_i,x_j,c)$, then $|t|_i = [x_{j+n}|c]$. and hence $\hat{\mathcal{N}_t} = \{[x_j|c]\}$. On the other hand $\mathcal{N}_t = \{[x_j|c]\}$, so again we have isomorphism.

Finally, if $t\in S_i$ and $\sigma(t) = (q_i,c,q_j)$, then $\mathcal{N}_t \cong c\cdot\mathcal{L}_{j,m}$. Since also $|t|_i = [x_j|c]$ we obtain $\hat{\mathcal{N}_t} \cong c\cdot \mathcal{L}_{j,m}$.

By 2.21 we conclude that $\mathcal{L}_i \cong \hat{\mathcal{L}}_i$. In particular $|D| = \hat{\mathcal{L}}_1 \cong \mathcal{L}_1 = \mathcal{L}_{\mathcal{A}} \cong \mathcal{L}$.

If $\mathcal{L}$ is recognizable, then $\mathcal{A}$ may be chosen without silent transitions. That is $S_i = \emptyset$ $(i=1,\dots,n)$. By construction $\mathcal{M}_i$ gets x_j-quasiregular $(i,j=1,\dots,n)$. Hence D is quasiregular in this case. $\square$

9.21 Corollary. *The iteration theory of weakly recognizable weighted tree-languages is equal to the smallest sub iteration theory of* $\mathrm{WTh}_{\Sigma(X)}$ *that contains all primitive arrows and that is closed with respect to equivalence* ($\equiv$). $\square$

9.22 Remark. The previous theorem is a Kleene-type result á la Bloom & Ésik (cf. [6, Cor. 10.3])—just for weakly recognizable weighted tree-languages. The smallest sub iteration theory of $\mathrm{WTh}_{\Sigma(X)}$ that contains all primitive arrows and that is closed with respect to equivalence is completely determined by its scalar arrows. The scalar arrows of this theory can be constructed as the smallest set of scalar arrows that contains all primitive scalar arrows and that is closed with respect to equivalence, $(-)^\dagger$ and scalar composition $F \cdot \langle F_1, \dots, F_n\rangle$ (where $F, F_1, \dots, F_n$ are scalar). This follows in particular from 8.13 where we noted that the general dagger-operation is determined by the scalar dagger-operation. This gives rise to a reformulation of 9.21 resulting in a Kleene-type result for weakly recognizable weighted tree-languages in the way of Bozapalidis [9]:

9.23 Corollary. *The class of weakly recognizable scalar morphisms in* $\mathrm{WTh}_{\Sigma(X)}$ *is equal to the smallest subclass of scalar morphisms from* $\mathrm{WTh}_{\Sigma(X)}$ *that contains all* $F_{\mathcal{L}}$ *for* $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X)}$ *finite and that is closed with respect to isomorphic copies, scalar composition scalar dagger.*

Proof. Let us denote the smallest subclass of scalar morphisms from $\mathrm{WTh}_{\Sigma(X)}$ that contains all $F_{\mathcal{L}}$ for $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X)}$ finite and that is closed with respect to isomorphic copies, scalar composition and scalar dagger by $\mathrm{WTh}^{\mathrm{rat}}_{\Sigma(X)}$.

The scalar composition and the scalar dagger preserve weak recognizability. Moreover, every finite weighted tree-language $\mathcal{L}$ is recognizable and hence so is $F_{\mathcal{L}}$. Therefore the elements of $\mathrm{WTh}^{\mathrm{rat}}_{\Sigma(X)}$ are indeed all weakly recognizable.

It remains to show the other inclusion. Note that for every primitive scalar arrow F we have that $\mathcal{L}(F)$ is finite and therefore $F \in \mathrm{WTh}^{\mathrm{rat}}_{\Sigma(X)}$. By the proof of 8.13 the dagger of each arrow can be computed from its scalar components using scalar composition and scalar dagger (cf. 8.14). Thus, given any primitive arrow $D = \langle D_1, \dots, D_{n+p}\rangle$, then each component of $D^\dagger = \langle T_1, \dots, T_n\rangle$ is in $\mathrm{WTh}^{\mathrm{rat}}_{\Sigma(X)}$.

Let now $F \in \mathrm{WTh}_{\Sigma(X_p)}$ be a weakly recognizable scalar morphism. Then by 9.20 there is a normal description $D = (\alpha, G) : 1 \longrightarrow p$ of weight n such that $F \equiv |D| = \alpha \cdot G^\dagger$. But by the argument from above $|D|$ is in $\mathrm{WTh}^{\mathrm{rat}}_{\Sigma(X)}$. Hence $F \in \mathrm{WTh}^{\mathrm{rat}}_{\Sigma(X)}$. $\square$

10 Fixed Point Theory of Formal Tree-Series

In this section we will define an algebraic theory of formal tree-series and study its properties. Such tree-series theories have already been studied by Bloom and Ésik in the special case that the coefficient-semiring is a commutative Conway- or iteration-semiring (cf. [6]). In this setting we always obtain a Conway- or iteration- grove theory of formal tree-series. We shall show that given a commutative coefficient-semiring K, the theory of formal tree-series is a grove-theory that is at the same time a partial iteration-theory (in case of a monadic ranked alphabet we can even drop the requirement of commutativity for K). We will use this result in particular to characterize recognizable formal tree-series as the behavior of so called normal descriptions. This shows that our weighted tree-automata have the same recognizing power as the normal descriptions—the automata-model used in fixed point theory (cf. [4, 6]). We go on describing the recognizable formal tree-series as components of the unique solution of proper linear systems of equations. Such a result was shown before by Berstel and Reutenauer [2] in the case when K is a field and by Ésik and Kuich in the case when K is a complete semiring [22].

Let us start with some further definitions from the field of algebraic theories. The notions follow closely [4] and [6].

10.1 Pointed theories. An algebraic theory T is called *pointed* if it contains a distinguished morphism $0 : 1 \longrightarrow 0$. If T is pointed then we may define morphisms $0_{n,p} : n \longrightarrow p$ according to

$$0_{n,p} := \overbrace{\langle 0, \ldots, 0\rangle}^{n \text{ times}} \cdot 0_p.$$

Thus $0 = 0_{1,0}$. A *pointed* T-*morphism* between pointed theories is a T-morphism that preserves the point 0.

10.2 Grove theories. Let T be a pointed theory. Assume that each hom-set $T(n,p)$ of T is equipped with a binary operation $+$. Then $(T, (i_j), 0, +)$ is called *grove-theory* if T is a theory and for all $n, p \in \mathbb{N}$ the hom-set $(T(n,p), +, 0_{n,p})$ forms a commutative monoid such that

1. $(f+g)\cdot h = f\cdot h + g\cdot h \quad$ for all $f, g : n \longrightarrow p$, $h : p \longrightarrow m$,
2. $0_{r,n}\cdot f = 0_{r,p} \quad$ for all $f : n \longrightarrow p$,
3. $i_n \cdot (f+g) = i_n\cdot f + i_n \cdot g \quad$ for all $f, g : n \longrightarrow p$, $i = 1, \ldots, n$,
4. $i_n \cdot 0_{n,p} = 0_{1,p} \quad$ for all $i = 1, \ldots, n$.

A homomorphism between grove-theories is a pointed T-morphism that preserves the monoid-structure of each hom-set.

10.3 Ideals in theories. Let T be an algebraic theory. A collection I of morphisms of T is called *ideal* of T if:

1. $\forall f_1, \ldots, f_n : 1 \longrightarrow p \quad f_1, \ldots, f_n \in I \Rightarrow \langle f_1, \ldots, f_n \rangle \in I$,
2. $\forall f : n \longrightarrow m,\ g : m \longrightarrow p \quad g \in I \Rightarrow f \cdot g \in I$,
3. $\forall f : n \longrightarrow m,\ g : m \longrightarrow p \quad f \in I, g \text{ base } \Rightarrow f \cdot g \in I$.

10.4 Partial pre-iteration theories. A *partial pre-iteration theory* is a triple $(T, I, \dagger)$ such that T is a theory, I is an ideal in T and $\dagger$ maps arrows $f : n \longrightarrow n+p$ from I to $f^\dagger : n \longrightarrow p$ in T ($n, p \in \mathbb{N}$). A homomorphism between partial pre-iteration theories is a T-morphism that preserves the ideal and that is compatible with the partial operation $\dagger$.

10.5 Partial iteration theories. A partial pre-iteration theory is called *partial iteration theory* if it satisfies the following (partial) identities:

1. Left zero identity
$$(0_n \oplus f)^\dagger = f$$
for all $f : n \longrightarrow p$ from I,
2. Base parameter identity
$$(f \cdot (\mathbf{1}_n \oplus g))^\dagger = f^\dagger \cdot g$$
for all $f : n \longrightarrow n + p$ from I, $g : p \longrightarrow q$ base,
3. Pairing identity
$$\langle f, g \rangle^\dagger = \langle f^\dagger \cdot \langle h^\dagger, \mathbf{1}_p \rangle, h^\dagger \rangle$$
for all $f : n \longrightarrow n + m + p$ from I, $g : m \longrightarrow n + m + p \in I$ where $h = g \cdot \langle f^\dagger, \mathbf{1}_{m+p} \rangle : m \longrightarrow m + p$,
4. Commutative identity
$$((\varrho \cdot f) \parallel (\varrho_1, \ldots, \varrho_m))^\dagger = \varrho \cdot (f \cdot (\varrho \oplus \mathbf{1}_p))^\dagger$$
for all $f : n \longrightarrow m + p$ from I, $\varrho : m \longrightarrow n$ surjective base, $\varrho_i : m \longrightarrow m$ base such that $\varrho_i \cdot \varrho = \varrho$ ($i = 1, \ldots, m$) (cf. 8.15).

10.6 OI-substitution. Next we introduce the operation of *OI-substitution* on formal tree-series. Let $t \in \mathrm{T}_{\Sigma(X_n)}$, $S_1, \dots, S_n \in \mathsf{FTS}_{\Sigma(X_m)}$. Then $t \cdot \langle S_1, \dots, S_n \rangle$ is defined by induction on the structure of T:

$$(x_i \cdot \langle S_1, \dots, S_n \rangle, s) := (S_i, s)$$

$$(a \cdot \langle S_1, \dots, S_n \rangle, s) := \begin{cases} 1 & s = a \\ 0 & \text{else} \end{cases}$$

$$(f\langle t_1, \dots, t_k \rangle \cdot \langle S_1, \dots, S_n \rangle, s) := ([f|1]\langle t_1 \cdot \langle S_1, \dots, S_n \rangle, \dots, t_k \cdot \langle S_1, \dots, S_n \rangle \rangle, s).$$

For $S \in \mathsf{FTS}_{\Sigma(X_n)}$ we define

$$(S \cdot \langle S_1, \dots, S_n \rangle, s) := \bigoplus_{t \in \mathrm{T}_{\Sigma(X_n)}} (S, t) \odot (t \cdot \langle S_1, \dots, S_n \rangle, s).$$

This operation is welldefined since for all but finitely many $t \in \mathrm{T}_{\Sigma(X_n)}$ we have $(t \cdot \langle S_1, \dots, S_n \rangle, s) = 0$.

10.7 Lemma. *Assume K is commutative. Let $u \in \mathrm{WT}_{\Sigma(X_n)}$ and let $\mathcal{L}_1, \dots, \mathcal{L}_n \in \mathsf{WTL}_{\Sigma(X_m)}$ be finitary. Then*

$$(S_{u \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n \rangle}, s) = \mathrm{wt}(u) \odot (\mathrm{ut}(u) \cdot \langle S_{\mathcal{L}_1}, \dots, S_{\mathcal{L}_n} \rangle, s)\,.$$

Proof. We proceed by induction on the structure of u.

If $u = [a|c]$, then

$$(S_{u \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n \rangle}, s) = (S_{\{[a|c]\}}, s) = \begin{cases} c & s = a \\ 0 & \text{else.} \end{cases}$$

and on the other hand

$$\begin{aligned} \mathrm{wt}(u) \odot (\mathrm{ut}(u) \cdot \langle S_{\mathcal{L}_1} \ \dots, S_{\mathcal{L}_n} \rangle, s) &= c \odot (a \cdot \langle S_{\mathcal{L}_1} \ \dots, S_{\mathcal{L}_n} \rangle, s) \\ &= c \odot (1a, s) = \begin{cases} c & s = a \\ 0 & \text{else.} \end{cases} \end{aligned}$$

If $u = [x_i|c]$ then

$$(S_{u \cdot \langle \mathcal{L}_1, \dots, \mathcal{L}_n \rangle}, s) = (S_{c \cdot \mathcal{L}_i}, s) = c \odot (S_{\mathcal{L}_i}, s)$$

and on the other hand

$$\mathrm{wt}(u) \odot (\mathrm{ut}(u) \cdot \langle S_{\mathcal{L}_1}, \dots, S_{\mathcal{L}_n} \rangle, s) = c \odot (x_i \cdot \langle S_{\mathcal{L}_1}, \dots, S_{\mathcal{L}_n} \rangle, s) = c \odot (S_{\mathcal{L}_i}, s).$$

If $u = [f|c]\langle u_1, \dots, u_k\rangle$, then

$$
\begin{aligned}
(S_{u\cdot\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle}, s) &= (S_{[f|c]\langle u_i\cdot\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle\rangle_{i=1}^k}, s)\\
&= ([f|c]\langle S_{u_1\cdot\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle}, \dots, S_{u_k\cdot\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle}\rangle, s)\\
&= \begin{cases} c \odot \bigodot_{i=1}^{k} (S_{s_i\cdot\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle}, s_i) & s = f\langle s_1, \dots, s_k\rangle\\ 0 & \text{else}\end{cases}\\
&= \begin{cases} c \odot \bigodot_{i=1}^{k} \mathrm{wt}(u_i) \odot (\mathrm{ut}(u_i)\cdot\langle S_{\mathcal{L}_1}, \dots, S_{\mathcal{L}_n}\rangle, s_i) & s = f\langle s_1, \dots, s_k\rangle\\ 0 & \text{else}\end{cases}\\
&\qquad \text{and since } K \text{ is commutative:}\\
&= \begin{cases} (c \odot \bigodot_{i=1}^{k} \mathrm{wt}(u_i)) \odot \bigodot_{i=1}^{k}(\mathrm{ut}(u_i)\cdot\langle S_{\mathcal{L}_1}, \dots, S_{\mathcal{L}_n}\rangle, s_i) & s = f\langle s_1, \dots, s_k\rangle\\ 0 & \text{else}\end{cases}\\
&= \mathrm{wt}(u) \odot (\mathrm{ut}(u)\cdot\langle S_{\mathcal{L}_1}, \dots, S_{\mathcal{L}_n}\rangle, s).
\end{aligned}
$$

□

10.8 Remark. Note that in the proof above we needed commutativity of K only for the case where $u = [f|c]\langle u_1, \dots, u_k\rangle$ and where $k > 1$. Hence, if Σ only contains letters of rank less than or equal 1, then 10.7 holds for arbitrary semirings K.

10.9 Lemma. *Assume K is commutative. Let $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X_n)}$ and let $\mathcal{L}_1, \dots, \mathcal{L}_n \in \mathsf{WTL}_{\Sigma(X_m)}$ be finitary. Then $S_{\mathcal{L}\cdot\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle} = S_{\mathcal{L}}\cdot\langle S_{\mathcal{L}_1}, \dots, S_{\mathcal{L}_n}\rangle$.*

Proof.

$$
\begin{aligned}
(S_{\mathcal{L}\cdot\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle}, s) &= \bigoplus_{\substack{u\in\mathcal{L}\cdot\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle\\ \mathrm{ut}(|u|)=s}} \mathrm{wt}(|u|)\\
&= \bigoplus_{u\in\mathcal{L}} \bigoplus_{\substack{v\in|u|\cdot\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle\\ \mathrm{ut}(|v|)=s}} \mathrm{wt}(|v|) = \bigoplus_{u\in\mathcal{L}} (S_{|u|\cdot\langle\mathcal{L}_1,\dots,\mathcal{L}_n\rangle}, s)\\
&= \bigoplus_{u\in\mathcal{L}} \mathrm{wt}(|u|) \odot (\mathrm{ut}(|u|)\cdot\langle S_{\mathcal{L}_1}, \dots, S_{\mathcal{L}_n}\rangle, s) \qquad \text{(by 10.7)}\\
&= \bigoplus_{t\in\mathrm{T}_{\Sigma(X_n)}} \left(\bigoplus_{\substack{u\in\mathcal{L}\\ \mathrm{ut}(|u|)=t}} \mathrm{wt}(|u|)\right) \odot (t\cdot\langle S_{\mathcal{L}_1}, \dots, S_{\mathcal{L}_n}\rangle, s) \quad \text{(by distr.)}\\
&= \bigoplus_{t\in\mathrm{T}_{\Sigma(X_n)}} (S_{\mathcal{L}}, t) \odot (t\cdot\langle S_{\mathcal{L}_1}, \dots, S_{\mathcal{L}_n}\rangle, s)\\
&= (S_{\mathcal{L}}\cdot\langle S_{\mathcal{L}_1}, \dots, S_{\mathcal{L}_n}\rangle, s).
\end{aligned}
$$

$\square$

10.10 Remark. Above, the requirement of commutativity was only necessary since we used 10.7. By 10.8 we can drop this requirement if Σ is monadic.

10.11 A theory of formal tree-series. Next we define the theory $\mathrm{FTh}_{\Sigma(X)}$ of formal tree-series. The scalar arrows $S : 1 \longrightarrow p$ are elements of $\mathsf{FTS}_{\Sigma(X_p)}$ and the arrows $T : n \longrightarrow p$ are n-tuples $\langle T_1, \dots, T_n \rangle$ of elements of $\mathsf{FTS}_{\Sigma(X_p)}$. For $n \in \mathbb{N}$ and $1 \leq i \leq n$, the distinguished arrow $i_n : 1 \longrightarrow n$ is the formal tree-series from $\mathsf{FTS}_{\Sigma(X_n)}$ that acts according to

$$i_n : t \mapsto \begin{cases} 1 & t = x_i \\ 0 & \text{else.} \end{cases}$$

For $S = \langle S_1, \dots, S_n \rangle : n \longrightarrow p$ and $t = \langle T_1, \dots, T_p \rangle : p \longrightarrow m$ we define

$$S \cdot T := \langle S_1 \cdot \langle T_1, \dots, T_p \rangle, \dots, S_n \cdot \langle T_1, \dots, T_p \rangle \rangle : n \longrightarrow m.$$

The actual proof that this defines indeed a theory will be postponed until 10.13.

10.12 Lemma. *Assume that K is commutative. Then $\mathrm{FTh}_{\Sigma(X)}$ may be obtained as epimorphic image of the theory $\mathrm{WTh}^{\mathrm{fin}}_{\Sigma(X)}$ of finitary weighted tree-languages (cf. 9.15). This epimorphism is induced by $F \mapsto S_{\mathcal{L}(F)}$ for $F : 1 \longrightarrow n$ finitary. The distinguished arrows are preserved. Moreover, if $F \equiv G$, then $S_{\mathcal{L}(F)} = S_{\mathcal{L}(G)}$.*

Proof. For $n \in \mathbb{N}$ and $1 \leq i \leq n$, the arrow $i_n \in \mathrm{WTh}_{\Sigma(X)}$ is $F_{\{[x_i|1]\}}$. Of course $\mathcal{L}(F) = \{[x_i|1]\}$ and $S_{\mathcal{L}(F)} : t \mapsto \begin{cases} 1 & t = x_i \\ 0 & \text{else} \end{cases}$. But this is i_n in $\mathrm{FTh}_{\Sigma(X)}$. Hence the distinguished arrows are preserved.

For $F : n \longrightarrow p$ and $G : p \longrightarrow m$ in $\mathrm{WTh}^{\mathrm{fin}}_{\Sigma(X)}$ we compute

$$\begin{aligned} S_{\mathcal{L}(F \cdot G)} &= S_{\langle \mathcal{L}_i(F) \cdot \langle \mathcal{L}_1(G), \dots, \mathcal{L}_p(G) \rangle \rangle_{i=1}^n} && \text{by 9.14} \\ &= \langle S_{\mathcal{L}_i(F) \cdot \langle \mathcal{L}_1(G), \dots, \mathcal{L}_p(G) \rangle} \rangle_{i=1}^n \\ &= \langle S_{\mathcal{L}_i(F)} \cdot \langle S_{\mathcal{L}_1(G)}, \dots, S_{\mathcal{L}_p(G)} \rangle \rangle_{i=1}^n && \text{by 10.9} \\ &= \langle S_{\mathcal{L}_1(F)}, \dots, S_{\mathcal{L}_n(F)} \rangle \cdot \langle S_{\mathcal{L}_1(G)}, \dots, S_{\mathcal{L}_p(G)} \rangle \\ &= S_{\mathcal{L}(F)} \cdot S_{\mathcal{L}(G)}. \end{aligned}$$

Hence the assignment $F \mapsto S_{\mathcal{L}(F)}$ is a T-morphism.

It remains to show surjectivity on the hom-sets. For this it is enough to prove it on scalar arrows. Let $\eta : \mathrm{T}_\Sigma \longrightarrow \mathrm{WT}_\Sigma$ be the canonical embedding which maps every tree t to the weighted tree $\hat{t}$ such that the underlying tree of $\hat{t}$ is t and all weights of $\hat{t}$ are equal to 1.[7] Define $\mathcal{L} := (\mathrm{T}_\Sigma, |.|)$ where $|t| := (S, t) \cdot \eta(t)$. Then $S_{\mathcal{L}} = S$. Hence we have surjectivity.

[7] this is indeed the free homomorphism of ranked monoids induced by the mapping $f \mapsto [f|1]$ where $f \in \Sigma(X)$; cf. also 1.21.

Let finally $F, G : n \longrightarrow p \in \mathrm{WTh}^{\mathrm{fin}}_{\Sigma(X)}$ such that $F \equiv G$. Then $\mathcal{L}(F) \cong \mathcal{L}(G)$ and hence $S_{\mathcal{L}(F)} = S_{\mathcal{L}(G)}$. □

10.13 Corollary. *Assume K is commutative. Then $\mathrm{FTh}_{\Sigma(X)}$ is indeed a theory. Moreover, for $S \in \mathsf{FTS}_{\Sigma(X_n)}$, $S_1, \ldots, S_n \in \mathsf{FTS}_{\Sigma(X_p)}$, $T_1, \ldots, T_p \in \mathsf{FTS}_{\Sigma(X_m)}$, $c \in K$, $f \in \Sigma^{(n)}$ we have:*

1. $(c \cdot S) \cdot \langle S_1, \ldots, S_n \rangle = c \cdot (S \cdot \langle S_1, \ldots, S_n \rangle)$,
2. $[f|c]\langle S_1, \ldots, S_n \rangle \cdot \langle T_1, \ldots, T_p \rangle = [f|c]\langle S_i \langle T_1, \ldots, T_p \rangle\rangle_{i=1}^n$,
3. $(S \cdot \langle S_1, \ldots, S_n \rangle) \cdot \langle T_1, \ldots, T_n \rangle = S \cdot \langle S_i \cdot \langle T_1, \ldots, T_p \rangle\rangle_{i=1}^n$,
4. $S \cdot \langle S_1, \ldots, S_n \rangle = (S \cdot_{x_1} x_{p+1} \cdots \cdot_{x_n} x_{p+n}) \cdot_{x_{p+1}} S_1 \cdots \cdot_{x_{p+n}} S_n$.

Proof. We start by proving 1, 2, 3 and 4:

Let $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X_n)}$, $\mathcal{L}_1, \ldots, \mathcal{L}_n \in \mathsf{WTL}_{\Sigma(X_p)}$, $\mathcal{M}_1, \ldots, \mathcal{M}_p \in \mathsf{WTL}_{\Sigma(X_m)}$ such that $S = S_{\mathcal{L}}$, $S_i = S_{\mathcal{L}_i}$ $(i = 1, \ldots, n)$, $T_i = S_{\mathcal{M}_i}$ $(i = 1, \ldots, p)$.

Ad 1:

$$
\begin{aligned}
(c \cdot S) \cdot \langle S_1, \ldots, S_n \rangle &= (c \cdot S_{\mathcal{L}}) \cdot \langle S_{\mathcal{L}_1}, \ldots, S_{\mathcal{L}_n} \rangle \\
&= S_{c \cdot \mathcal{L}} \cdot \langle S_{\mathcal{L}_1}, \ldots, S_{\mathcal{L}_n} \rangle && \text{by } 7.12 \\
&= S_{(c \cdot \mathcal{L}) \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n \rangle} && \text{by } 10.9 \\
&= S_{c \cdot (\mathcal{L} \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n \rangle)} && \text{by } 9.5 \\
&= c \cdot S_{\mathcal{L} \cdot \langle \mathcal{L}_1, \ldots, \mathcal{L}_n \rangle} && \text{by } 7.12 \\
&= c \cdot (\mathcal{S}_{\mathcal{L}} \cdot \langle S_{\mathcal{L}_1}, \ldots, S_{\mathcal{L}_n} \rangle) && \text{by } 10.9 \\
&= c \cdot (S \cdot \langle S_1, \ldots, S_n \rangle).
\end{aligned}
$$

Ad 2:

$$
\begin{aligned}
([f|c]\langle S_1, \ldots, S_n \rangle) \cdot \langle T_1, \ldots, T_p \rangle &= ([f|c]\langle S_{\mathcal{L}_1}, \ldots, S_{\mathcal{L}_n} \rangle) \cdot \langle S_{\mathcal{M}_1}, \ldots, S_{\mathcal{M}_p} \rangle \\
&= (S_{[f|c]\langle \mathcal{L}_1, \ldots, \mathcal{L}_n \rangle}) \cdot \langle S_{\mathcal{M}_1}, \ldots, S_{\mathcal{M}_p} \rangle && \text{by } 7.16 \\
&= S_{([f|c]\langle \mathcal{L}_1, \ldots, \mathcal{L}_n \rangle) \cdot \langle \mathcal{M}_1, \ldots, \mathcal{M}_p \rangle} && \text{by } 10.9 \\
&= S_{[f|c]\langle \mathcal{L}_i \cdot \langle \mathcal{M}_1, \ldots, \mathcal{M}_p \rangle\rangle_{i=1}^n} && \text{by } 9.5 \\
&= [f|c]\langle S_{\mathcal{L}_i \cdot \langle \mathcal{M}_1, \ldots, \mathcal{M}_p \rangle}\rangle_{i=1}^n && \text{by } 7.16 \\
&= [f|c]\langle S_{\mathcal{L}_i} \cdot \langle S_{\mathcal{M}_1}, \ldots, S_{\mathcal{M}_p} \rangle\rangle_{i=1}^n && \text{by } 10.9 \\
&= [f|c]\langle S_i \cdot \langle T_1, \ldots, T_p \rangle\rangle_{i=1}^n
\end{aligned}
$$

Ad 3:

$$
\begin{aligned}
(S \cdot \langle S_1, \ldots, S_n \rangle) \cdot \langle T_1, \ldots, T_p \rangle &= (S_{\mathcal{L}} \cdot \langle S_{\mathcal{L}_1}, \ldots, S_{\mathcal{L}_n} \rangle) \cdot \langle S_{\mathcal{M}_1}, \ldots, S_{\mathcal{M}_p} \rangle \\
&= S_{(\mathcal{L} \cdot \mathcal{L}_1, \ldots, \mathcal{L}_n \rangle) \cdot \langle \mathcal{M}_1, \ldots, \mathcal{M}_p \rangle} && \text{by } 10.9 \\
&= S_{\mathcal{L} \cdot \langle \mathcal{L}_i \cdot \langle \mathcal{M}_1, \ldots, \mathcal{M}_p \rangle\rangle_{i=1}^n} && \text{by } 9.5 \\
&= S_{\mathcal{L}} \cdot \langle S_{\mathcal{L}_i} \cdot \langle S_{\mathcal{M}_1}, \ldots, S_{\mathcal{M}_p} \rangle\rangle_{i=1}^n && \text{by } 10.9 \\
&= S \cdot \langle S_i \cdot \langle T_1, \ldots, T_p \rangle\rangle_{i=1}^n.
\end{aligned}
$$

Ad 4:

$$\begin{aligned}
S\cdot\langle S_1,\ldots,S_n\rangle &= S_{\mathcal{L}}\cdot\langle S_{\mathcal{L}_1},\ldots,S_{\mathcal{L}_n}\rangle \\
&= S_{\mathcal{L}\cdot\langle\mathcal{L}_1,\ldots,\mathcal{L}_n\rangle} && \text{by 10.9} \\
&= S_{(\mathcal{L}\cdot_{x_1}\{[x_{p+1}|1]\}\cdots\cdot_{x_n}\{[x_{p+n}|1]\})\cdot_{x_{p+1}}\mathcal{L}_1\cdots\cdot_{x_{p+n}}\mathcal{L}_n} && \text{by 9.6} \\
&= (S_{\mathcal{L}}\cdot_{x_1} S_{\{[x_{p+1}|1]\}}\cdots\cdot_{x_n} S_{\{[x_{p+n}|1]\}})\cdot_{x_{p+1}} S_{\mathcal{L}_1}\cdots\cdot_{x_{p+n}} S_{\mathcal{L}_n} && \text{by 7.14} \\
&= (S\cdot_{x_1} x_{p+1}\cdots\cdot_{x_n} x_{p+n})\cdot_{x_{p+1}} S_1\cdots\cdot_{x_{p+n}} S_n
\end{aligned}$$

From point 3 follows already that the composition of arrows in $\mathrm{FTh}_{\Sigma(X)}$ is associative. It remains to show the existence of unit-arrows for each object and the universal property of the distinguished morphisms (cf. 8.1). Set $\mathbf{1}_n := \langle 1_, \ldots, n_n\rangle$ in $\mathrm{FTh}_{\Sigma(X)}$. Then $\mathbf{1}_n = S_{\mathcal{L}(\mathbf{1}_n)}$ because distinguished arrows are preserved by $S_{\mathcal{L}(-)}$ and since $\mathbf{1}_n = \langle 1_n, \ldots, n_n\rangle$ in $\mathrm{WTh}_{\Sigma(X)}$. Let $S : n \longrightarrow p$ be a morphism in $\mathrm{FTh}_{\Sigma(X)}$. Then $S = S_{\mathcal{L}(F)}$ for some $F : n \longrightarrow p$ in $\mathrm{WTh}_{\Sigma(X)}$. But then

$$\begin{aligned}
\mathbf{1}_n\cdot S &= S_{\mathcal{L}(\mathbf{1}_n)}\cdot S_{\mathcal{L}(F)} \\
&= S_{\mathcal{L}(\mathbf{1}_n\cdot F)} \\
&= S_{\mathcal{L}(F)} = S.
\end{aligned}$$

Moreover, let $T : p \longrightarrow n$ from $\mathrm{FTh}_{\Sigma(X)}$. Then $T = S_{\mathcal{L}(G)}$ for some $G : p \longrightarrow n$ from $\mathrm{WTh}_{\Sigma(X)}$. Then we compute

$$\begin{aligned}
T\cdot\mathbf{1}_n &= S_{\mathcal{L}(G)}\cdot S_{\mathcal{L}(\mathbf{1}_n)} \\
&= S_{\mathcal{L}(G\cdot\mathbf{1}_n)} \\
&= S_{\mathcal{L}(G)} = T.
\end{aligned}$$

Hence $\mathbf{1}_n$ is indeed a unit-morphisms of n in $\mathrm{FTh}_{\Sigma(X)}$. The fact that that from $T\cdot\langle S_1,\ldots,S_n\rangle = S_i$ for all $S_1,\ldots,S_n : n \longrightarrow p$ from $\mathrm{FTh}_{\Sigma(X)}$ follows that $T = i_n$ can be shown by taking in particular $S_i = i_n$ $(i = 1,\ldots,n)$. In this case

$$T = T\cdot\mathbf{1}_n = T\cdot\langle 1_n,\ldots,n_n\rangle = i_n.$$

Thus $\mathrm{FTh}_{\Sigma(X)}$ is a theory. $\square$

10.14 Corollary. *Assume K is commutative. Then for $S, S' \in \mathsf{FTS}_{\Sigma(X_n)}$, and for $S_1,\ldots,S_n \in \mathsf{FTS}_{\Sigma(X_p)}$ we have that*

$$(S+S')\cdot\langle S_1,\ldots,S_n\rangle = S\cdot\langle S_1,\ldots,S_n\rangle + S'\cdot\langle S_1,\ldots,S_n\rangle.$$

Proof. Let $\mathcal{L},\mathcal{L}' \in \mathsf{WTL}_{\Sigma(X_n)}$, $\mathcal{L}_1,\ldots,\mathcal{L}_n \in \mathsf{WTL}_{\Sigma(X_p)}$ such that $S = S_{\mathcal{L}}$, $S' = S_{\mathcal{L}'}$

and $S_i = S_{\mathcal{L}_i}$ $(i = 1, \dots, n)$. Then

$$
\begin{aligned}
(S + S') \cdot \langle S_1, \dots, S_n \rangle &= (S_{\mathcal{L}} + S_{\mathcal{L}'}) \cdot \langle S_{\mathcal{L}_1}, \dots, S_{\mathcal{L}_n} \rangle \\
&= S_{\mathcal{L}+\mathcal{L}'} \cdot \langle S_{\mathcal{L}_1}, \dots, S_{\mathcal{L}_n} \rangle && \text{by 7.10} \\
&= S_{(\mathcal{L}+\mathcal{L}')\cdot\langle \mathcal{L}_1, \dots, \mathcal{L}_n \rangle} && \text{by 10.9} \\
&= S_{\mathcal{L}\cdot\langle \mathcal{L}_1, \dots, \mathcal{L}_n \rangle + \mathcal{L}'\cdot\langle \mathcal{L}_1, \dots, \mathcal{L}_n \rangle} && \text{by 9.4} \\
&= S_{\mathcal{L}\cdot\langle \mathcal{L}_1, \dots, \mathcal{L}_n \rangle} + S_{\mathcal{L}'\cdot\langle \mathcal{L}_1, \dots, \mathcal{L}_n \rangle} && \text{by 7.10} \\
&= S_{\mathcal{L}} \cdot \langle S_{\mathcal{L}_1}, \dots, S_{\mathcal{L}_n} \rangle + S_{\mathcal{L}'} \cdot \langle S_{\mathcal{L}_1}, \dots, S_{\mathcal{L}_n} \rangle && \text{by 10.9} \\
&= S \cdot \langle S_1, \dots, S_n \rangle + S' \cdot \langle S_1, \dots, S_n \rangle.
\end{aligned}
$$

□

10.15 Remark. We turn $\mathrm{FTh}_{\Sigma(X)}$ into a pointed theory by choosing the point $0 : 1 \longrightarrow 0$ as the series from FTS_Σ that maps each tree to the zero-element of K. Clearly $0_{1,n}$ is the corresponding 0-series of $\mathsf{FTS}_{\Sigma(X_n)}$ and $0_{n,p} = \overbrace{\langle 0_{1,p}, \dots, 0_{1,p} \rangle}^{n \text{ times}}$. On the hom-set $\mathrm{FTh}_{\Sigma(X)}(n, p)$ we may introduce an operation '+' according to

$$S + T := \langle S_1, \dots, S_n \rangle + \langle T_1, \dots, T_n \rangle := \langle S_1 + T_1, \dots, S_n + T_n \rangle.$$

10.16 Proposition. (Bloom, Ésik [6]) $(\mathrm{FTh}_{\Sigma(X)}, +, 0)$ *is a grove-theory.*

Proof. It is clear that the hom-sets form commutative monoids with the given operations. So it remains to prove that the four axioms for grove-theories are fulfilled.

Ad 1: This is an immediate consequence of 10.14.

Ad 2: Since $0_{r,n} = \overbrace{\langle 0_{1,n}, \dots, 0_{1,n} \rangle}^{r \text{ times}}$ and $0_{1,n}$ is the 0-series of $\mathsf{FTS}_{\Sigma(X_n)}$, we compute for $\langle S_1, \dots, S_n \rangle : n \longrightarrow p$:

$$
\begin{aligned}
\langle 0_{1,n}, \dots, 0_{1,n} \rangle \cdot \langle S_1, \dots, S_n \rangle &= \langle 0_{1,n} \cdot \langle S_1, \dots, S_n \rangle, \dots, 0_{1,n} \cdot \langle S_1, \dots, S_n \rangle \rangle \\
&= \langle 0_{1,p}, \dots, 0_{1.p} \rangle \\
&= 0_{r,p}.
\end{aligned}
$$

Ad 3: For $\langle S_1, \dots, S_n \rangle, \langle T_1, \dots, T_n \rangle : n \longrightarrow p$, $1 \le i \le n$ we compute

$$
\begin{aligned}
i_n \cdot (\langle S_1, \dots, S_n \rangle \cdot \langle T_1, \dots, T_n \rangle) &= i_n \cdot \langle S_1 + T_1, \dots, S_n + T_n \rangle \\
&= S_i + T_i \\
&= i_n \cdot \langle S_1, \dots, S_n \rangle + i_n \cdot \langle T_1, \dots, T_n \rangle.
\end{aligned}
$$

Ad 4: $i_n \cdot 0_{n,p} = 0_{1,p}$ by definition of $0_{n,p}$ (cf. 10.11). □

10.17 Recognizability, full quasiregularity, quasiregularity. Like in Section 9 we may associate attributes of formal tree-series to arrows of $\mathrm{FTh}_{\Sigma(X)}$. A morphism $S = \langle S_1, \ldots, S_n \rangle : n \longrightarrow p$ is called *recognizable* if S_i is recognizable for all $1 \leq i \leq n$. It is called *fully quasiregular* if S_i is x_j-quasiregular for all $1 \leq i \leq n$ and $1 \leq j \leq p$. An arrow $T = \langle T_1, \ldots, T_n \rangle : n \longrightarrow n + p$ is called *quasiregular* if T_i is x_j-quasiregular for $1 \leq i, j \leq n$. The collection of fully quasiregular and quasiregular morphisms of $\mathrm{FTh}_{\Sigma(X)}$ will be denoted by I and J, respectively.

10.18 Lemma. *The collection I of all fully quasiregular morphisms of* $\mathrm{FTh}_{\Sigma(X)}$ *forms an ideal.*

Proof. Obviously fully quasiregular morphisms are closed with respect to source-tupling. Let $S : n \longrightarrow p \in I$ and $T : p \longrightarrow m$. Then

$$S \cdot T = \langle S_1, \ldots, S_n \rangle \cdot \langle T_1, \ldots, T_p \rangle = \langle S_i \cdot \langle T_1, \ldots, T_p \rangle \rangle_{i=1}^{n}.$$

Now we compute for $1 \leq j \leq m$

$$(S_i \cdot \langle T_1, \ldots, T_p \rangle, x_j) = \bigoplus_{t \in T_{\Sigma(X)}} (S_i, t) \odot (t \cdot \langle T_1, \ldots, T_p \rangle, x_j).$$

But $(t \cdot \langle T_1, \ldots, T_p \rangle, x_j) \neq 0$ implies $t = x_k$ for some $1 \leq k \leq p$ and $(T_k, x_j) \neq 0$. But in this case $(S_i, x_k) = 0$ and therefore $(S_i \cdot \langle T_1, \ldots, T_p \rangle, x_j) = 0$ for all $1 \leq j \leq p$ and hence it is fully quasiregular. Thus $S \cdot T$ is fully quasiregular.

If $T : m \longrightarrow n$ is base, then $T = \langle (i_1)_n, \ldots, (i_m)_n \rangle$ for certain $i_1, \ldots, i_m \in \{1, \ldots, n\}$. Hence $T \cdot S = \langle S_{i_1}, \ldots, S_{i_m} \rangle$ which is obviously fully quasiregular again. $\square$

10.19 Dagger-operation on $\mathbf{FTh}_{\Sigma(\mathbf{X})}$. The previous lemma allows us to introduce a †-operation on J in $\mathrm{FTh}_{\Sigma(X)}$. In particular we define for $S : n \longrightarrow n + p$ from J:

$$S^\dagger := S_{F^\dagger} \qquad \text{for any } F : n \longrightarrow n + p \text{ in } \mathrm{WTh}^{\mathrm{fin}}_{\Sigma(X)} \text{ with } S_F = S.$$

Such an F clearly exists (cf. 10.12). The next few paragraphs show that it is also welldefined.

10.20 Lemma. *Let $S : 1 \longrightarrow 1 + p$ be from* $\mathrm{FTh}_{\Sigma(X)}$ *(that is $S \in \mathsf{FTS}_{\Sigma(X_{1+p})}$). Suppose that S is quasiregular. Then $S^\dagger = S^{\mu}_{x_1} \cdot \langle 1_p, \mathbf{1}_p \rangle$.*

Proof. Let $F : 1 \longrightarrow 1 + p$ be from $\mathrm{WTh}_{\Sigma(X)}$ such that $S_{\mathcal{L}(F)} = S$ (in particular F is quasiregular, hence $\mathcal{L}(F)$ is x_1-quasiregular). Then

$$\begin{aligned}
S^\dagger &= S_{\mathcal{L}(F^\dagger)} \\
&= S_{\mathcal{L}(F_{\mathcal{L}(F)^{\mu}_{x_1}} \cdot \langle 1_p, \mathbf{1}_p \rangle)} && \text{by 9.10 and 9.11} \\
&= S_{\mathcal{L}(F_{\mathcal{L}(F)^{\mu}_{x_1}})} \cdot \langle 1_p, \mathbf{1}_p \rangle && \text{by 10.9} \\
&= S_{\mathcal{L}(F)^{\mu}_{x_1}} \cdot \langle 1_p, \mathbf{1}_p \rangle && \text{see below} \\
&= S^{\mu}_{x_1} \cdot \langle 1_p, \mathbf{1}_p \rangle && \text{by 7.22}
\end{aligned}$$

where we also compute that

$$\begin{aligned}\mathcal{L}(F_{\mathcal{L}(F)^{\mu}_{x_1}}) &= F_{\mathcal{L}(F)^{\mu}_{x_1}}(\{[x_1|1]\},\dots,\{[x_{p+1}|1]\}) && \text{cf. 9.7}\\ &= \mathcal{L}(F)^{\mu}_{x_1}\cdot\langle\{[x_1|1]\},\dots,\{[x_{p+1}|1]\}\rangle\\ &\cong \mathcal{L}(F)^{\mu}_{x_1}.\end{aligned}$$

□

10.21 Lemma. *Let* $F_1, F_2 : n \longrightarrow n+p$ *in* $\mathrm{WTh}^{\mathrm{fin}}_{\Sigma(X)}$ *be quasiregular. Then* $S_{\mathcal{L}(F_1)} = S_{\mathcal{L}(F_2)} \Rightarrow S_{\mathcal{L}(F_1^\dagger)} = S_{\mathcal{L}(F_2^\dagger)}$.

Proof. We proceed by induction on n and use the pairing-identity:

Suppose that $F_1,\ F_2 : 1 \longrightarrow 1+p \in \mathrm{WTh}^{\mathrm{fin}}_{\Sigma(X)}$ such that $S_{\mathcal{L}(F_1)} = S_{\mathcal{L}(F_2)}$. Then we have by 10.20:

$$\begin{aligned}S_{\mathcal{L}(F_1^\dagger)} &= (S_{\mathcal{L}(F_1)})^{\mu}_{x_1}\cdot\langle 1_p, \mathbf{1}_p\rangle\\ &= (S_{\mathcal{L}(F_2)})^{\mu}_{x_1}\cdot\langle 1_p, \mathbf{1}_p\rangle\\ &= S_{\mathcal{L}(F_2^\dagger)}.\end{aligned}$$

Suppose now $F_1,\ F_2 : n+1 \longrightarrow n+1+p \in \mathrm{WTh}^{\mathrm{fin}}_{\Sigma(X)}$ such that $S_{\mathcal{L}(F_1)} = S_{\mathcal{L}(F_2)}$. Then for $i \in \{1,2\}$ we have $F_i = \langle \hat{F}_i, F_i'\rangle$ where $\hat{F}_i : 1 \longrightarrow n+1+p$, $F_i' : n \longrightarrow n+1+p$. By the paring identity $F_i^\dagger \cong \langle \hat{F}_i^\dagger \cdot \langle H_i^\dagger, \mathbf{1}_p\rangle, H_i^\dagger\rangle$ where $H_i = F_i'\cdot\langle \hat{F}_i^\dagger, \mathbf{1}_{n+p}\rangle : n \longrightarrow n+p$. Since $S_{\mathcal{L}(F_1)} = S_{\mathcal{L}(F_2)}$, we also have $S_{\mathcal{L}(\hat{F}_1)} = S_{\mathcal{L}(\hat{F}_2)}$ and $S_{\mathcal{L}(F_1')} = S_{\mathcal{L}(F_2')}$. By induction hypothesis $S_{\mathcal{L}(\hat{F}_1^\dagger)} = S_{\mathcal{L}(\hat{F}_2^\dagger)}$. Hence, by 10.12, $S_{\mathcal{L}(H_1)} = S_{\mathcal{L}(H_2)}$. Again by induction hypothesis $S_{\mathcal{L}(H_1^\dagger)} = S_{\mathcal{L}(H_2^\dagger)}$ and again by 10.12 we conclude $S_{\mathcal{L}(F_1^\dagger)} = S_{\mathcal{L}(F_2^\dagger)}$. □

10.22 Corollary. *The dagger-operation from 10.19 is welldefined.* □

10.23 Proposition. *Let* lhs = rhs *be any identity (cf. 8.18) that holds in* $\mathrm{WTh}_{\Sigma(X)}$ *up to equivalence. Then for any valuation* V *of the variable symbols in* $\mathrm{FTh}_{\Sigma(X)}$*, if* $[\mathrm{lhs}]_V$ *and* $[\mathrm{rhs}]_V$ *are defined, then they are equal.*

Proof. The mapping $F \mapsto S_{\mathcal{L}(F)}$ is a surjective T-morphism. Restricted to quasiregular elements of $\mathrm{WTh}^{\mathrm{fin}}_{\Sigma(X)}$ this T-morphism preserves the dagger-operation † (by definition of † in $\mathrm{FTh}_{\Sigma(X)}$). Let lhs = rhs be any identity that holds in $\mathrm{WTh}_{\Sigma(X)}$ up to equivalence. Let V be any valuation of the variable symbols in $\mathrm{FTh}_{\Sigma(X)}$ (cf. 8.17). Because of the surjectivity of the T-morphism $S_{\mathcal{L}(-)}$, there exists a valuation V' in $\mathrm{WTh}_{\Sigma(X)}$ such that for any variable-symbol $f : n \longrightarrow m$ we have that $S_{\mathcal{L}(V_{n,m}(f))} = V'_{n,m}(f)$. Since in $\mathrm{WTh}_{\Sigma(X)}$ the identity lhs = rhs holds up to equivalence, we have that $[\mathrm{lhs}]_{V'} \equiv [\mathrm{rhs}]_{V'}$. Hence, if $[\mathrm{lhs}]_V$ and $[\mathrm{rhs}]_V$ are defined in $\mathrm{FTh}_{\Sigma(X)}$, then $[\mathrm{lhs}]_{V'}, [\mathrm{rhs}]_{V'}$ are finitary and by 10.12 and since $S_{\mathcal{L}(-)}$ preserves the dagger-operation on quasiregular arrows, we get that

$$[\mathrm{lhs}]_V = S_{\mathcal{L}([\mathrm{lhs}]_{V'})} = S_{\mathcal{L}([\mathrm{rhs}]_{V'})} = [\mathrm{rhs}]_V.$$

□

10.24 Theorem. $(\mathrm{FTh}_{\Sigma(X)}, I, \dagger)$ *is a partial iteration theory.*

Proof. We have to show that in $(\mathrm{FTh}_{\Sigma(X)}, I, \dagger)$ the left zero identity, the base parameter identity, the pairing identity and the commutative identity hold for morphisms from I. Since the mentioned identities hold in $\mathrm{WTh}_{\Sigma(X)}$, it is enough to show that the respective left- and right-hand sides are defined for all proper valuations with morphisms from I.

Claim 1: $(0_n \oplus f)^\dagger = f$:
Consider $\mathrm{T}_{\Sigma(X)}$ as the free Σ-algebra freely generated by X. Let the endomorphism $\eta_n : \mathrm{T}_{\Sigma(X)} \longrightarrow \mathrm{T}_{\Sigma(X)}$ be induced by $x_i \mapsto x_{i+n}$ $(i \in \mathbb{N})$. Let now $S_f : n \longrightarrow p$ from I. Then

$$(0_n \oplus S_f, s) = \begin{cases} (S_f, t) & \eta_n(t) = s \\ 0 & \text{else.} \end{cases}$$

It is obvious that $0_n + S_f$ is again fully quasiregular. Hence both, the left-hand-side (lhs) and the right-hand-side (rhs) of the identity are defined.

Claim 2: $(f \cdot (\mathbf{1} \oplus g))^\dagger = f^\dagger \cdot g$:
Let $S_f \in I$, $S_g \in \mathrm{FTh}_{\Sigma(X)}$ base. Then $S_f \cdot (\mathbf{1}_n \oplus S_g) \in I$. Hence lhs and rhs are defined.

Claim 3: $\langle f, g\rangle^\dagger = \langle f^\dagger \cdot \langle h^\dagger, \mathbf{1}_p\rangle, h^\dagger\rangle$:
Let $S_f, S_g \in I$. Then $\langle S_f, S_g\rangle \in I$. Hence lhs is defined. From $S_g \in I$ follows that $S_h = S_g \cdot \langle S_f^\dagger, \mathbf{1}_{m+p}\rangle \in I$. Hence rhs is defined as well.

Claim 4: $((\varrho \cdot f) \parallel (\varrho_1, \ldots, \varrho_m))^\dagger = \varrho \cdot (f \cdot (\varrho \oplus \mathbf{1}_p))^\dagger$:
Let $S_f \in I$ and let $S_\varrho \in \mathrm{FTh}_{\Sigma(X)}$ surjective base and $S_{\varrho_i} \in \mathrm{FTh}_{\Sigma(X)}$ base such that $S_{\varrho_i} \cdot S_\varrho = S_\varrho$ $(i = 1, \ldots, m)$. Since $S_f \in I$ and S_ϱ base we have $S_\varrho \cdot S_f \in I$. With $S_\varrho \cdot S_f = \langle S_1, \ldots, S_m\rangle$ we have $S_i \in I$ and hence $S_i \cdot (S_{\varrho_i} \oplus \mathbf{1}_p) \in I$. Hence by 8.15, the lhs is defined.

Since $S_f \in I$ we see that $S_f \cdot (S_\varrho \oplus \mathbf{1}_p) \in I$. Hence rhs is defined as well. □

10.25 Proposition. *Let* $S = \langle S_1, \ldots, S_n\rangle : n \longrightarrow n + p$ *be quasiregular. Suppose* $S^\dagger = \langle T_1, \ldots, T_n\rangle$. *Then for all* $i = 1, \ldots, n$ *we have* $T_i \in \mathrm{Rat}(S_1, \ldots, S_n)$.

Proof. This is a direct consequence of 9.16 and of the fact that the mapping $\mathcal{L} \mapsto S_\mathcal{L} : \mathsf{WTL}^{\mathrm{fin}}_{\Sigma(X)} \longrightarrow \mathsf{FTS}_{\Sigma(X)}$ is surjective and preserves sum, product with scalars, x-product and x-recursion (restricted to x-quasiregular $\mathcal{L}$) (cf. 7.10, 7.12 and 7.22). □

10.26 Corollary. *If* $S : n \longrightarrow n + p$ *is quasiregular and recognizable, then* $S^\dagger$ *is recognizable.* □

10.27 Normal descriptions in $\mathbf{FTh}_{\Sigma(\mathbf{X})}$. Next we are going to carry the results from 9.19 and 9.20 over to formal tree-series.

A tree is called *primitive* if $t = f\langle x_{i_1}, \dots, x_{i_n}\rangle$ for some $f \in \Sigma(X)^{(n)}$, $n \in \mathbb{N}$, $x_{i_j} \in X$, $j = 1, \dots, n$. Clearly, primitive trees are precisely the underlying trees of primitive weighted trees.

A formal tree-series is called *primitive* if its support is finite and consists only of primitive trees. Again it is clear that for each primitive formal tree-series S there is a primitive weighted tree-language $\mathcal{L}$ such that $S = S_{\mathcal{L}}$.

A morphism $T = \langle T_1, \dots, T_n\rangle : q \longrightarrow n + p$ from $\mathrm{FTh}_{\Sigma(X)}$ is called *primitive of weight* n if and only if T_i is primitive for all $i = 1, \dots, n$. Again it is easy to see that T is primitive of weight n if and only if there is a primitive morphism $F : q \longrightarrow n + p$ in $\mathrm{WTh}_{\Sigma(X)}$ of weight n such that $T = S_{\mathcal{L}(F)}$.

A *normal description* $D : k \longrightarrow p$ in $\mathrm{FTh}_{\Sigma(X)}$ of weight n is a pair $D = (\alpha, S)$ such that $S : n \longrightarrow n + p$ is quasiregular and primitive of weight n and $\alpha : k \longrightarrow n$ is base. The behavior of D is the morphism $|D| := \alpha \cdot S^\dagger$. Clearly, $D = (\alpha, S) : k \longrightarrow p$ is a normal description of weight n if and only if there is a quasiregular normal description $\hat{D} = (\beta, F) : k \longrightarrow p$ of weight n in $\mathrm{WTh}_{\Sigma(X)}$ such that $\alpha = S_{\mathcal{L}(\beta)}$ and $S = S_{\mathcal{L}(F)}$. Moreover, in this case $|D| = S_{\mathcal{L}(|\hat{D}|)}$.

10.28 Theorem. *Assume that K is commutative. For each normal description $D : k \longrightarrow p$ in $\mathrm{FTh}_{\Sigma(X)}$ we have that $|D|$ is recognizable. Moreover, if $S \in \mathsf{FTS}_{\Sigma(X)}$ is recognizable, then there is a normal description $D : 1 \longrightarrow p$ in $\mathrm{FTh}_{\Sigma(X)}$ such that $S = |D|$.*

Proof. Let $D = (\alpha, T) : k \longrightarrow p$ be a normal description. Then there is a quasiregular normal description $\hat{D} = (\beta, F) : k \longrightarrow p$ in $\mathrm{WTh}_{\Sigma(X)}$ such that $\alpha = S_{\mathcal{L}(\beta)}$ and $T = S_{\mathcal{L}(F)}$. By 9.19 $|\hat{D}| = \beta \cdot F^\dagger$ is recognizable. Hence, by definition, $S_{\mathcal{L}(|\hat{D}|)}$ is recognizable. Finally, by 10.9,

$$|D| = \alpha \cdot T^\dagger = S_{\mathcal{L}(\beta)} \cdot S_{\mathcal{L}(F^\dagger)} = S_{\mathcal{L}(\beta \cdot F^\dagger)} = S_{\mathcal{L}(|\hat{D}|)}.$$

Now let $S \in \mathsf{FTS}_{\Sigma(X_p)}$ be recognizable. Then, by definition, there is a recognizable weighted tree-language $\mathcal{L} \in \mathsf{WTL}_{\Sigma(X_p)}$ with $S = S_{\mathcal{L}}$. By 9.20 there is a quasiregular normal description $\hat{D} = (\beta, F) : 1 \longrightarrow p$ in $\mathrm{WTh}_{\Sigma(X)}$ such that $F_{\mathcal{L}} \equiv |\hat{D}|$. Define $D := (\alpha, T) : 1 \longrightarrow p$ in $\mathrm{FTh}_{\Sigma(X)}$ according to $\alpha := S_{\mathcal{L}(\beta)}$ and $T := S_{\mathcal{L}(F)}$. Obviously D is quasiregular. Then

$$\begin{aligned} |D| = \alpha \cdot T^\dagger &= S_{\mathcal{L}(\beta)} \cdot (S_{\mathcal{L}(F)})^\dagger \\ &= S_{\mathcal{L}(\beta)} \cdot S_{\mathcal{L}(F^\dagger)} \\ &= S_{\mathcal{L}(\beta \cdot F^\dagger)} = S_{\mathcal{L}(|\hat{D}|)} = S. \end{aligned}$$

□

10.29 Remark. The previous theorem in conjunction with 7.25 generalizes the Kleene-type theorem for formal tree-series by Bloom and Ésik from [6]. In section 10 of this paper they remark that in case if K is a (commutative) Conway-semiring the rational series coincide with the behaviours of normal descriptions. They obtain this result from a more general result about Conway grove-theories (cf. [6, Theorem 9.4]).

10.30 Remark. Let $S : n \longrightarrow n+p$ be quasiregular. Then, by the fixed point identity (which holds by 10.23), we have $S^\dagger = S \cdot \langle S^\dagger, \mathbf{1}_p\rangle$. That is $S^\dagger$ is a solution to the fixed point equation $Y = S \cdot \langle Y, \mathbf{1}_p\rangle$.

10.31 Theorem. *Let K be commutative and let $S : n \longrightarrow n+p$ in $\mathrm{FTh}_{\Sigma(X)}$ be quasiregular. Then $S^\dagger$ is the only solution to the fixed point equation $Y = S\cdot\langle Y, \mathbf{1}_p\rangle$.*

Proof. The idea of the following proof is due to Berstel and Reutenauer who proved a weaker result on formal power-series over fields in [2].

Assume $Y = \langle Y_1, \dots, Y_n\rangle$ is a solution. We will show that each value (Y_i, t) there is just one choice and that (Y_i, t) depends only on coefficients of S_j and the values (Y_j, u) for $j = 1, \dots, n$ with $\mathrm{depth}(u) < \mathrm{depth}(t)$. This will be done by induction on the depth of t.

$\mathrm{depth}(t) = 0$: If $t = x_j$ $(j = 1, \dots, p)$, then

$$(Y_i, t) = (Y_i, x_j) = (S_i \cdot \langle Y_1, \dots, Y_n, 1_p, \dots, p_p\rangle, x_j) = (S_i, x_{n+j})$$

since S_i is quasiregular. For the same reason, if $t = a \in \Sigma(X)^{(0)}$, then

$$(Y_i, t) = (Y_i, a) = (S_i \cdot \langle Y_1, \dots, Y_n, 1_p, \dots, p_p\rangle, x_j) = (S_i, a).$$

$\mathrm{depth}(t) = n+1$: Suppose $t = f\langle t_1, \dots, t_k\rangle$ with $\mathrm{depth}(t_i) \le n$. Then

$$\begin{aligned}(Y_i, t) &= (S_i \cdot \langle Y_1, \dots, Y_n, 1_p, \dots, p_p\rangle, f\langle t_1, \dots, t_k\rangle)\\ &= \bigoplus_{s\in \mathrm{T}_{\Sigma(X_p)}} (S_i, s) \odot (s \cdot \langle Y_1, \dots, Y_n, 1_p, \dots, p_p\rangle, f\langle t_1, \dots, t_k\rangle)\end{aligned}$$

For $s \in \mathrm{supp}(S_i)$ define $Z_s := s \cdot \langle Y_1, \dots, Y_n, 1_p, \dots, p_p\rangle$. It remains to show that for the value $(Z_s, f\langle t_1, \dots, t_k\rangle)$ depends only on the coefficients of the Y_j on trees of depth $\le n$. Since S_i is quasiregular, we always have $s \ne x_j$ for $j = 1, \dots, n$. Hence $f\langle t_1, \dots, t_n\rangle \in \mathrm{supp}(Z_s)$ implies $s = f\langle s_1, \dots, s_k\rangle$ for certain trees $s_1, \dots, s_n$. But then it is clear that

$$\begin{aligned}(Z_s, f\langle t_1, \dots, t_k\rangle) &= ([f|1]\langle s_j \cdot \langle Y_1, \dots, Y_n, 1_p, \dots, p_p\rangle\rangle_{j=1}^k, f\langle t_1, \dots, t_k\rangle)\\ &= \bigodot_{j=1}^{k} (s_j \cdot \langle Y_1, \dots, Y_n, 1_p, \dots, p_p\rangle, t_j).\end{aligned}$$

However, this depends only on the values (Y_j, t_l) for $j = 1, \dots, n$ and $l = 1, \dots, k$ which are given by induction hypothesis. □

10.32 Remark. Note that the previous theorem does not only show the uniqueness of the solution of $Y = S \cdot \langle Y, \mathbf{1}_p\rangle$ but also its existence. Hence we could have defined $S^\dagger$ immediately to be the unique fixed point of this equation. However, in this case it would be rather difficult to show the many identities that hold for this dagger-operation. With our method we obtain the validity of those identities almost for free. The close connection of the dagger-operation to the x_1-recursion (cf. 9.10) gives another argument for the naturality of the iteration-operations we chose.

10.33 Remark about polynomial systems. A morphism $S : n \longrightarrow n+p$ is called *polynomial* if each of its components is a polynomial. Polynomial systems $S : n \longrightarrow n$ were studied by Berstel and Reutenauer [2]. What they called proper systems, we call quasiregular morphisms from n to n. They showed that their proper systems have precisely one solution and that this solution has only recognizable components and that moreover each recognizable formal tree-series appears as a component of a solution of a proper system. Above we generalized these results to formal tree-series over commutative semirings. For us systems of equations are particular morphisms of $\mathrm{FTh}_{\Sigma(X)}$, namely from n to $n+p$. The existence of solutions for quasiregular systems (morphisms) follows from 10.19 and 10.23. The uniqueness of the solution for quasiregular systems (and in principle again the existence) was shown in 10.31. The fact that quasiregular polynomial systems have a recognizable solution follows from 10.26 and from the easy fact that polynomials are recognizable. Note that normal descriptions are in principle special quasiregular polynomial systems. Thus, by 10.28, each recognizable formal power series appears as a component of a solution of a quasiregular polynomial system.

10.34 Corollary. *Assume that K is commutative. The set of all recognizable scalar arrows from $\mathrm{FTh}_{\Sigma(X)}$ is equal to the smallest subset of scalar arrows of $\mathrm{FTh}_{\Sigma(X)}$ that contains all polynomials and that is closed with respect to scalar composition and scalar dagger (restricted to quasiregular arrows).*

Proof. Let us denote the smallest subset of scalar arrows from $\mathrm{FTh}_{\Sigma(X)}$ that contains all polynomials and that is closed with respect to OI-substitution and x-recursion by $\mathrm{FTh}^{\mathrm{rat}}_{\Sigma(X)}$. First we note that all elements of $\mathrm{FTh}^{\mathrm{rat}}_{\Sigma(X)}$ are recognizable because all polynomials are recognizable and because scalar composition and scalar dagger preserve recognizability.

For the other inclusion we use 10.28. Let S be some recognizable scalar morphism from $\mathrm{FTh}_{\Sigma(X)}$ then there is a normal description $D = (\alpha, T) : 1 \longrightarrow p$ of weight n with $S = |D| = \alpha \cdot T^\dagger$. Note that each of the components of $T : \langle T_1, \ldots, T_n + p \rangle$ is a polynomial. Using the same argument as in the proof of 9.23, we obtain that the scalar components of $T^\dagger$ belong to $\mathrm{FTh}^{\mathrm{rat}}_{\Sigma(X)}$. Hence also $|D| \in \mathsf{FTS}^{\mathrm{rat}}_{\Sigma(X)} \in \mathrm{FTh}^{\mathrm{rat}}_{\Sigma(X)}$. $\square$

10.35 Remark. The previous result may be seen as a generalization of the Kleene-type theorem by Bozapalidis in [9] which assumes a commutative well ω-additive coefficient-semiring. It assumes K to be commutative. In general, many of our results in this section hold only under the assumption that the coefficient-semiring K is commutative. Note however, that whenever Σ is a monadic ranked alphabet, then we can drop the assumption of commutativity.

11 Concluding Remarks

11.1 The general strategy. When analyzing the available literature containing Kleene-type results for formal tree-series, it became apparent that the restrictions these papers impose on the coefficient semiring become necessary because too many computations are carried out too early directly in the semiring. For instance [36, 22, 6] use fixed point theory. For this it is needed that the Lawvere theory of formal tree-series admits a fixed point operation that additionally satisfies many axioms. This works for instance if the semiring carries a fixed point operation itself (if for instance the semiring is complete, continuous or a Conway-semiring). The restriction in [18] to idempotent semirings has a similar reason. Only then is their iteration operation welldefined.

It was our thesis from the beginning that Kleene-type results for formal tree-series should be essentially independent of the semiring. Our method of resolution was to give a semantics to weighted automata for which the structure of the semiring is of no concern—this is, for its definition as few calculations in the semiring as possible should be necessary. In this way the results obtained eventually become independent of the algebraic structure of the semiring.

To reach this goal there are different possibilities. One extremal example is the final semantics. Thereby the WTAs are understood as coalgebras (cf. e.g. [32, 29, 46]). The semantics of a state is then its image in the final coalgebra. Two states are called bisimilar if the have the same final semantics. It is easy to see that the final semantics is adequate— that is, two bisimilar states define equal formal tree-series. On the other hand this approach is technically too demanding. There are easier ways to reach the goal.

The WTL-semantics that we chose lies in between the final and the FTS-semantics. This means that any two bisimilar states define isomorphic weighted tree-languages and isomorphic weighted tree-languages define equal formal tree-series. The advantage of WTL-semantics over final semantics is that weighted tree-languages are easier to handle than elements of the final weighted automaton. In fact weighted tree-languages are structurally very similar to formal tree-languages. A further algebraic motivation is, that the set of weighted Σ-trees is nothing but the free weighted Σ-algebra. This makes the induction-principle to the main proof principle in our work.

11.2 The definition of weighted tree-languages. Weighted tree-languages can be seen as multisets. We modeled them as a carrier-set together with a structure map $|.|$ that assigns a weighted tree to each element of the carrier. Intuitively we would like to identify each element of the carrier with its weighted tree. Thus a weighted tree may be in the weighted tree-language several times. With this perception of multisets our concept of weighted tree-languages comes very close to the concept of formal tree-languages. One advantage is that operations on formal tree-languages can be generalized easily to weighted tree-languages. Another advantage is that on multisets of this kind there is very naturally a notion of functions and, related to this, a notion of homomorphisms. There a homomorphism between multisets is just a function between the carriers that preserves the structure-map. It is very essential to note that the whole (!) algebraic structure on weighted tree-languages such as products, coproducts, union, intersection is determined by this homomorphism-concept. Indeed, all these operations arise as certain limit- or colimit constructions. As a last advantage we note that this kind of weighted tree-languages supports very well automata-theoretic constructions, since the semantics of the state of an automaton can be taken as the weighted tree-language whose carrier-set is the set of runs of the automaton starting in this state and where the structure map on the runs is defined naturally. The compatibility of all the automata-constructions with the respective rational operations can then be proved by finding an isomorphism between the respective weighted tree-languages. There is of course a price to pay. Working with the machinery of category-theory is considered to be overly abstract and technical by many mathematicians. However, we think that the advantages outweigh this inconvenience. Moreover we only use very basic terminology from category theory and all our categorial constructions have a nice set-theoretical intuition in background.

Another way to define weighted tree-languages would have been the combinatorial way. A weighted tree-language in this sense is a function from WT_Σ to $\mathbb{N} \cup \{\infty\}$. It assigns to each weighted tree its multiplicity in the multiset. In this way the weighted tree-languages form a complete lattice (with respect to pointwise order) with smallest element 0 (the function that maps everything to 0) and with largest element ∞ (the function that maps everything to ∞). It carries another semiring-structure according to pointwise addition and pointwise multiplication. There the additive unit is 0 and the multiplicative unit is 1 (the function that maps everything to 1). It can be shown then that addition and multiplication are in fact continuous operations with respect to the complete lattice-structure, so indeed we have a complete continuous semiring of weighted tree-languages in this case. This approach has indisputable advantages if we would like to examine weighted tree-languages just with fixed point theoretical methods. Indeed it can be shown that weighted tree-languages form a grove-theory that is at the same time an iteration theory. The fixed point theoretical Kleene-type results can then be obtained as consequence of a general Kleene-type theorem for grove-theories that are Conway-theories [6]. However, we have quite a few objections about this way. First the operations of addition, multiplication, infimum and supremum do not come naturally with this definition of weighted tree-languages but they have to be imposed artificially. This is in contrast with our approach where such constructions arise as a consequence of a natural homomorphism concept. Secondly we think that the classical Kleene-theorem lives greatly from the intuitive simplicity of its rational operations. The definitions of the rational operations on this kind of weighted tree-languages would involve many computations in $\mathbb{N} \cup \{\infty\}$. The iteration-operations would be constructed by some supremum-construction. This is algebraically all very nice but it is not immediately clear what the operations do to the languages and how they generalize the rational operations on formal tree-languages. Also the fixed point theoretical proofs are by no means elementary but they root on results about fixed point identities in complete partial orders or iterative theories that are by no means trivial.

11.3 Weighted tree-automata. Our definition of weighted tree-automata differs in one point from the usual definitions. In particular the (silent) transitions of our automata form finite multisets. While this would not be absolutely necessary, it simplifies our automata-constructions. For instance the construction of the a-recursion introduces new silent transitions. If we did not have a multiset of silent transitions it could happen that the new transitions "overwrite" already existing transitions. But this leads to an unexpected result.

11.4 Rational and fp-expressions. It is customary to call expressions that are formed using operations on formal power-series or formal tree-series "rational expressions". This term roots in the algebraic theory of power-series over fields. There holds $S^* = \frac{1}{1-S}$ for all proper formal power-series. So rational expressions in this case can indeed be formed using the well known rational operations of addition, multiplication and division. It is our opinion that the expressions that can be formed from addition, multiplication with scalars, topcatenation and x-recursion, are not a proper generalization of the known concept of rational expressions. In particular the x-recursion does not generalize the Kleene-star on formal power series because the specialization of the x-recursion to formal power-series maps every series to the trivial power-series 0. Derived from the close relation of the x-recursion with the fixed point operation on the theory of formal tree-series, we have called such expressions "fixed point expressions".

We always used the term "rational expression" when the respective iteration operation was either x-iteration or x-semiiteration. Indeed, the x-iteration generalizes the Kleene-star and the x-semiiteration generalizes the Kleene-plus operation where $S^+ = \frac{S}{1-S} = SS^*$ for a formal power-series S.

11.5 Fixed point theory. It was our goal to generalize automata-theoretic and fixed point theoretic Kleene-type results from [36, 9, 22, 6, 18]. To obtain a satisfying solution we first proved our Kleene-type result automata-theoretical. This way it was immediately clear that we generalized the result by Droste and Vogler [18]. As the next step we had to conclude the fixed-point theoretic results from our automata-theoretic result. To this end we developed a bit the fixed point theory of weighted tree-languages. While doing so the above mentioned combinatorial definition of weighted tree-languages would have been more convenient. Within our categorial notion of weighted tree-languages we have to live with some inconveniences. For instance the Lawvere-theory of weighted tree-languages is not small. Moreover the fixed point identities do not hold there exactly but only up to natural isomorphism. Some results we even only prove up to equivalence of functors (two functors F_1, F_2 are equivalent if $\forall X : F_1(X) \cong F_2(X)$). But this is of no big importance since the fixed point theory on weighted tree-languages is just auxiliary for the fixed point theory of formal tree-series, which does not suffer from such problems.

11.6 Schützenberger's theorem. Let Σ be a monadic ranked alphabet. That is all elements except for one letter $*$ have rank 1 and $*$ has rank 0. Then the set T_Σ of trees consists essentially of formal words over the (non ranked) alphabet $\Sigma \setminus \{*\}$. It is also clear that the recognizable formal tree-series over Σ are in a one-to-one correspondence with the recognizable formal power-series over $\Sigma \setminus \{*\}$. Let S be a recognizable formal power series over $\Sigma\setminus\{*\}$ and let $\hat{S}$ be its corresponding formal tree-series. By 7.25 there is an fp-expression defining $\hat{S}$. Unfortunately this fp-expression involves certain variable symbols such that subexpressions define in general formal tree-series over $\Sigma(X_n)$ (recall that $X_n = \{x_1, \dots, x_n\}$). On the other hand by Schützenberger's Theorem there is a rational expression r that defines S. However, in these rational expressions there do not appear any variable symbols. Note that any formal tree-series $\hat{S}$ over $\Sigma(X_n)$ is well described by a tuple $(S_0, S_1, \dots, S_n)$ of formal power-series where S_0 describes the weights of monadic trees ending with $*$ and S_i describes the weights of monadic trees ending with x_i. Now we will sketch how to translate an fp-expression e with variables only from X_n defining $\hat{S}$ into a tuple $[\![e]\!] = (r_0, r_1, \dots, r_n)$ such that r_i defines S_i $(i = 0, \dots, n)$. Of course this is done by induction on the structure of e: $[\![*]\!] := (1, 0, \dots, 0)$, $[\![x_i]\!] := (\underbrace{0, 0, \dots, 0}_{i \text{ times}}, 1, 0, \dots, 0)$, $[\![f\langle e_1\rangle]\!] := (f \cdot r_0, \dots, f \cdot r_n)$ where $[\![e_1]\!] = (r_0, \dots, r_n)$, $[\![e_1 + e_2]\!] := (r_{1,0} + r_{2,0}, \dots, r_{1,n} + r_{2,n})$ where $[\![r_i]\!] = (r_{i,0}, \dots, r_{i,n})$ $(i = 1, 2)$, $[\![\mu x_i.e]\!] := (r_i^* r_0, \dots, r_i^* r_{i-1}, 0, r_i^* r_{i+1}, \dots, r_i^* r_n)$ where $[\![e]\!] = (r_0, \dots, r_n)$. If e is a closed fp-expression then it defines a formal tree-series $\hat{S}$ over Σ. Now if $[\![e]\!] = (r_0, \dots, r_n)$, then r_0 is a rational expression that defines the formal power-series S corresponding to $\hat{S}$. Hence, since for monadic alphabets we can always abandon the requirement of commutativity to the semiring, we can derive Schützenberger's theorem from our results.

11.7 Prospect.

A. Weighted tree-languages are multisets. If the structure map $|.|$ of a language $\mathcal{L} = (L, |.|)$ is injective, then $\mathcal{L}$ is essentially a set. Such languages we call definite. Now a recognizable formal tree-series may be called definite if if it is induced by a definite recognizable weighted tree-language. The first question is whether every recognizable formal tree-series is definite. If no, then the next question is whether definiteness is decidable.

B. Another problem field for which weighted tree-languages seem to be well-suited is the theory of algebraic formal tree-series. Here it should be possible to generalize results by Kuich [37] to a wider class of semirings.

C. When we define the cumulative weight of a weighted tree, we just multiply all weights from the nodes of the tree. With this we associate then a formal tree-series to the weighted tree-language and show that this assignment preserves the rational operations on formal tree-series. There are other ways to define the cumulative weight of a weighted tree. For instance the single weights of the weighted trees could be given different priorities according to their depth in the tree. For formal power series something like this was done by Droste and Kuske [17]. Their result can probably be generalized to formal tree-series.

D. Closely related to C. is the question about formal tree-series over infinite trees and their characterization by ω-rational expressions. Here it would be possible to work with infinite weighted trees and to prove for them the classical theorems and then to move the results to formal tree-series over infinite trees. A starting point could be the paper by Droste and Kuske [17].

E. The field of tree-series transducers (cf. [21, 24, 25]) could perhaps also benefit from the use of weighted tree-languages because the transducer-types known from the theory of formal tree-languages probably have natural generalizations to weighted tree-languages. In this realm they could be studied independently from the semiring and like in our Kleene-type result, the restrictions to the semiring would only occur when translating these results to formal tree-series. Thus the necessity of the respective restrictions would become more vivid.

References

[1] J. Adámek, H. Herrlich, and G. E. Strecker. *Abstract and Concrete Categories.* John Wiley & Sons, New York, 1990.

[2] J. Berstel and C. Reutenauer. Recognizable formal power series on trees. *Theoretical Computer Science*, 18:115–148, 1982.

[3] J. Berstel and C. Reutenauer. *Rational Series and Their Languages*, volume 12 of *EATCS Monographs*. Springer Verlag, 1988.

[4] S. L. Bloom and Z. Ésik. *Iteration Theories: The Equational Logic of Iterative Processes*. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1993.

[5] S. L. Bloom and Z. Ésik. The equational logic of fixed points. *Theoretical Computer Science*, 179:1–60, 1997.

[6] S. L. Bloom and Z. Ésik. An extension theorem with an application to formal tree series. *Journal of Automata, Languages and Combinatorics*, 8(2):145–185, 2003.

[7] F. Borceux. *Handbook of Categorical Algebra. 1. Basic category theory*, volume 50 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1994.

[8] B. Borchardt and H. Vogler. Determinization of finite state weighted tree automata. *Journal of Automata, Languages and Combinatorics*, 8(3):417–463, 2003.

[9] S. Bozapalidis. Equational elements in additive algebras. *Theory Comput. Systems*, 32:1–33, 1999.

[10] J. A. Brzozowski. Derivatives of regular expressions. *Journal of the ACM*, 11:481–494, 1964.

[11] A. L. Buchsbaum, R. Giancarlo, and J. R. Westbrook. On the determinization of weighted finite automata. *SIAM Journal on Computing*, 30(5):1502–1531, 2000.

[12] S. Burris and H. P. Sankappanavar. *A Course in Universal Algebra.* Web-publication, 2000. http://www.thoralf.uwaterloo.ca/htdocs/ualg.html.

[13] G. Cohen, S. Gaubert, and J.-P. Quadrat. Max-plus algebra and system theory: where we are and where to go now. *Annual Reviews in Control*, 23:207–219, 1999.

[14] K. Culik and J. Kari. Image compression using weighted finite automata. *Computer and Graphics*, 17(3):305–313, 1993.

[15] R. A. Cuninghame-Green. Minimax algebra and applications. *Advances in Imaging and Electron Physics*, 90:1–121, 1995.

[16] M. Droste and P. Gastin. The kleene-schützenberger theorem for formal power series in partially commuting variables. *Information and Computation*, 153:47–80, 1999. extended abstract in: *24th ICALP*, LNCS vol 1256, Springer, 1997, pp. 682–692.

[17] M. Droste and D. Kuske. Skew and infinitary formal power series. In *Automata Languages and Programming (30th ICALP, Eindhoven)*, volume 2719 of *Lecture Notes in Computer Science*, pages 426–438, 2003.

[18] M. Droste and H. Vogler. A Kleene theorem for weighted tree automata. technical report TUD-FI02-04, Technische Universität Dresden, June 2002.

[19] S. Eilenberg. *Automata, Languages, and Machines*, volume A. Academic Press, 1974.

[20] J. Engelfriet. Tree automata and tree grammars. Technical Report DAIMI FN-10, Aarhus University, 1975.

[21] J. Engelfriet, Z. Fülöp, and H. Vogler. Bottom-up and top-down tree series transformations. *J. Automata, Languages, and Combinatorics*, 7:11–70, 2002.

[22] Z. Ésik and W. Kuich. Formal tree series. *Journal of Automata, Languages and Combinatorics*, 8(2):219–285, 2003.

[23] Z. Ésik and A. Labella. Equational properties of iteration in algebraically complete categories. *Theoretical Computer Science*, 195:61–89, 1998.

[24] Z. Fülöp and H. Vogler. Tree series transformations that respect copying. *Theory of Computing Systems*, 36(3):247–293, 2003.

[25] Z. Fülöp and H. Vogler. Weighted tree transducers. Technical Report TUD-FI03-01, Dresden University of Technology, Jan. 2003.

[26] S. Gaubert and M. Plus. Methods and applications of $(max, +)$ linear algebra. Rapport de recherche 3088, INRIA, France, Jan. 1997.

[27] F. Gécseg and M. Steinby. Tree languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, chapter 1, pages 1–68. Springer Verlag, 1997.

[28] G. Grätzer. *Universal Algebra*. Springer Verlag, New York, 2nd edition, 1979.

[29] H. P. Gumm. Elements of the general theory of coalgebras. In *LUATCS'99*, Johannesburg, South Africa, 1999. Rand Africaans University.

[30] J. Gunawardena, editor. *Idempotency*. Cambridge University Press, Jan. 1998.

[31] U. Hafner. *Low Bit-Rate Image and Video Coding with Weighted Finite Automata*. Dissertation, Universität Würzburg, Germany, 1999.

[32] B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction. *Bulletin of the European Association for Theoretical Computer Science*, 62:222–259, 1997.

[33] Z. Jiang, B. Litow, and O. de Vel. Similarity enrichment in image compression through weighted finite automata. In X. Lin et al., editors, *COCOON'2000, Computing and Combinatorics*, volume 1858 of *Lecture Notes in Computer Science*, pages 447–456. Springer, 2000.

[34] F. Katritzke. *Refinements of data compression using weighted finite automata*. Dissertation, Universität Siegen, Germany, 2001.

[35] S. E. Kleene. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 3–42. Princeton University Press, Princeton, N.J., 1956.

[36] W. Kuich. Formal power series over trees. In *Proc. of the 3rd International Conference Developments in Language Theory*, pages 60–101, Aristotle University of Thesaloniki, 1997.

[37] W. Kuich. Pushdown tree automata, algebraic tree systems and algebraic tree series. *Information and Computation*, 165:1–31, 2001.

[38] W. Kuich and A. Salomaa. *Semirings, Automata, Languages*. EATCS Monographs on Theoretical Computer Science. Springer Verlag, 1986.

[39] A. Kurz. Limits in categories of coalgebras. a short note on a way to construct limits in categories of coalgebras. Draft, Nov. 1999.

[40] F. W. Lawvere. Functorial semantics of algebraic theories. In *Proceedings of the National Academy of Sciences USA*, volume 50, pages 869–873, 1963.

[41] S. Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer Verlag, New York, 2nd edition, 1998.

[42] M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311, 1997.

[43] M. Mohri, F. Pereira, and M. Riley. The design prinziples of a weighted finite-state transducer library. *Theoretical Computer Science*, 231:17–32, 2000.

[44] R. Pöschel and L. A. Kalužnin. *Funktionen- und Relationenalgebren*. VEB Deutscher Verlag der Wissenschaften, Berlin. Birkhäuser Verlag, Basel u. Stuttgart (Math. Reihe Bd. 67), 1979.

[45] W. Reisig. Über die Darstellung von Baumautomaten mittels Petrinetzen. unpublished manuscript.

[46] J. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.

[47] M. P. Schützenberger. On the definition of a family of automata. *Information and Control*, 4:245–270, 1961.

[48] W. Taylor. Abstract clone theory. In I. G. Rosenberg and G. Sabidussi, editors, *Algebras and Orders*, volume 389 of *Nato ASI Series, Series C: Mathematical and Physical Sciences*, pages 507–530. Kluwer Academic Publishers, 1993.

[49] J. W. Thatcher and J. B. Wright. Generalized finite automata theory with application to a decision problem of second-order logic. *Math. Systems Theory*, 2(1):57–81, 1968.

Versicherung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Die vorliegende Dissertation wurde an der Technischen Universität Dresden unter der Betreuung von Prof. Dr. Reinhard Pöschel angefertigt.

Affirmation

Hereby I affirm that I wrote the present thesis without any inadmissible help by a third party and without using any other means than indicated. Thoughts that were taken over directly or indirectly from other sources are indicated as such. This thesis has not been presented to any other examination board in this or a similar form, neither in this nor in any other country.

The present thesis was written at Dresden University of Technology under the supervision of Prof. Dr. Reinhard Pöschel.

Dresden, 27. März 2003